100% AMIGA

JAY MINER INTERVIEW

Amazing COMPUTING™ For The Commodore AMIGA

Your Original AMIGA® Monthly Resource

Volume 4 No. 9 September 1989
US $3.95 Canada $4.95

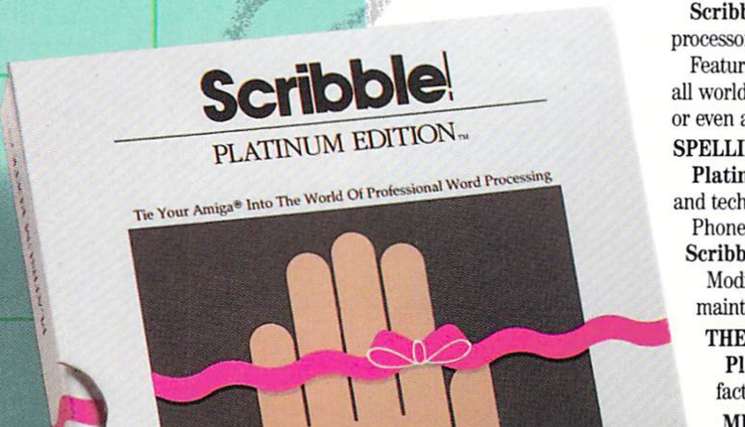
AMIGA GRAPHICS

AmiExpo Chicago '89

09
0 74470 74710 9

You're never too old to Scribble!

Scribble! Platinum Edition


Scribble!
PLATINUM EDITION™
Tie Your Amiga® Into The World Of Professional Word Processing
For The Amiga
MSS
better than ever

Scribble! Platinum Edition offers many improvements that make **Scribble!** the word processor of choice for beginning and experienced users.

Featuring rapid text entry and scrolling, **Scribble! Platinum Edition** offers the best of all worlds: power, speed, easy to use. In no time at all, you'll be writing memos, letters, or even a book!

SPELLING CHECKER

Platinum Scribble! shines in this area with a 104,000+ word dictionary with scientific and technical supplements. The Spell-As-You-Type option checks spelling on the fly.

Phonetic spelling, transposed letters, and contractions are no problem for **Platinum Scribble!** Enter *Kasil* and castle is suggested, *hte* finds the and *shalln't* finds shan't.

Modify the user dictionary by adding, editing and deleting words with the user dictionary maintenance program.

THESAURUS

Platinum Scribble! has a 470,000+ word thesaurus, with definitions. As a matter of fact, the word *right* displays 21 groups of synonyms.

MULTIPLE WINDOWS

Open up to 4 windows at once. Display your documents in 2, 4, or 8 color windows. Change the colors to suit your needs. **Platinum Scribble!** supports both interlace and non-interlace screens with horizontal and vertical overscan.

CUT AND PASTE

Copy, cut and paste text within a single document or from one document to another.

MAIL MERGE

Print form letters and mailing labels. Enter information at print time or automatically insert information from any mail merge text file.

IFF GRAPHIC SUPPORT

Print IFF graphics with any compatible printer supported by the Commodore Amiga printer drivers.

CLIPBOARD COMPATIBLE

Platinum Scribble! is clipboard compatible with Micro-Systems Software products.

OVERVIEW

• On-screen **bold**, *italic,* and underline, with highlighting for super and sub-script. • Support for laser printer font cartridges • Needs only 512k of memory • Not-copy protected • Easy hard drive installation • Free Technical Support for Registered Users.

MSS
Micro-Systems Software

Committed to excellence since 1978
12798 Forest Hill Boulevard, Suite 202
West Palm Beach, Florida 33414
407-790-0770 FAX 407-790-1341

Dealers and Distributors Call 1-800-327-8724

See your local dealer for a demonstration.

Scribble! Platinum Edition is a trademark of Micro-Systems Software, Inc. Amiga is a registered trademark of Commodore Business Machines.

Circle 101 on Reader Service card.

A3001

BREAKING THE SPEED BARRIER

New 25Mhz 68030 Upgrade Kit from GVP Puts Tomorrow's Performance in Your A2000 Today

SPEED 25Mhz 68030 CPU and 68882 FPU with fully implemented BURST mode 32-bit memory. An order of magnitude faster than A2500.

POWER Up to 8MB of State-of-the-art NIBBLE MODE 32-bit DRAM memory averages zero waitstates during 68030 BURST mode.

EFFICIENCY Built-in Hard Disk Controller direct on 32-bit bus keeps ALL A2000 expansion slots free even when our hard disk is installed!

QUALITY Backed up by GVP's unique full one year warranty.

A3001, 25Mhz 68030/68882 Upgrade Kit Technical Highlights:

- Asynchronous 68030 design allows high clock rates and GENLOCK compatibility.
- Factory installed 25Mhz 68882 floating point processor, twice as fast as the older 68881 chip.
- Fully DMA-able AUTOCON-FIGured 32-bit wide memory. Up to 8MB.
- Built-in AUTOBOOTing hard disk controller supports up to two 40 or 80MB drives (11/19ms access).
- Selectable 68000 fall-back mode for full floppy-based game compatibility.

Amiga is a registered trademark of Commodore-Amiga Inc.
IMPACT and GVP are trademarks of Great Valley Products, Inc.

GVP
GREAT VALLEY PRODUCTS INC.

For more information, or for your nearest GVP dealer, call today. Dealer inquiries welcome.

FAX (215) 889-9416 • (215) 889-9411 • BBS (215) 889-4994

225 Plank Ave., Paoli, PA 19301

Circle 158 on Reader Service card

Amazing COMPUTING™
Your Original AMIGA® Monthly Resource
For The Commodore AMIGA®

AMAZING COLUMNS

New Products...and Other Neat Stuff 15
by Elizabeth G. Fedorzyn
Muxic-X, Jack Nicklaus Golf, Scannery, and more!

HardCopy 44
by M. J. Bernier
Answers to AC 4.8 HardCopy.

Roomers 67
by the Bandito
The Bandito shares the untold tale of Amiga LIVE!

Snapshot 71
by Brad Andrews
Indiana Jones and the Temple of Doom, Rampage, BattleTech and more!

Bug Bytes 75
by John Steiner
Bugs in Superbase, upgrades for ARexx, and more!

Video Schmideo 80
by Barry Solomon
The Layman's guide to Amiga video!

C Notes from the C Group 86
by Stephen Kemp
"Keep it Simple" utility programming.

GRAPHICS & ANIMATION!

Secrets of Visual Synthesis 18
by David Duberman
Ray Tracing on the Amiga with Sculpt-Animate 4D and Turbo Silver 3.0.

PageRender 3D Review 23
by Michael Fahrion
Michael reviews Mindware's 3D rendering package.

DeluxePaint III — The Inside Story 27
by Ben & Jean Means
EA's Dan Silva tells how DeluxePaint III evolved.

Scene Generator Review 33
by R. Shamms Mortier, PhD.
A quick and easy-to-use graphics program—and it's free!

Fractal Presentation 35
by David Hiestand
David shares his presentation experiences.

Design-3D Review 39
by R. Shamms Mortier, PhD.
Mr. Mortier reviews Gold Disk's Design-3D.

Your Original AMIGA Monthly Resource

•TABLE OF CONTENTS•

Volume 4, Number 9
September, 1989

GRAPHICS & ANIMATION!

Digitizing Color Slides And Negatives on the Amiga 45
by Ron S. Gull
A basic guide to macro digitizing!

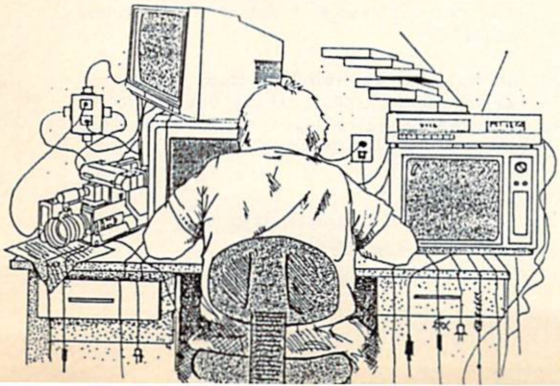
High Octane Colors 47
by Robert D'Asto
Maximizing your colors in AmigaBASIC.

Pixels At An Exhibition: The Video 50
by Shamms R. Mortier
Amiga-generated art is showcased at a University of Vermont show.

Cell Animation In Modula-2 62
by Nicholas Cirasella
How to walk all over your Amiga.

Improving Your Graphics Programming 89
by Richard Martin
Richard has the cure for the "Blind Artist's Syndrome."

Gels In Multi-Forth, Part III 94
by John Bushakra
John continues his series in Gels along with a tasty dish!



AMAZING FEATURES

T-Shell 41
by Rich Falconburg
A UNIX equivalent command line interface for communicating with your Amiga.

AmiEXPO '89 55
by AC's Managing Editor
Highlights from the Chicago '89 AmiEXPO!

Jay Miner Interview 58
by Steve Gillmor
An interview with the "Father of the Amiga".

Memory Squares 82
by Mike Morrison
Test your memory with Mike's memory game!

AMAZING DEPARTMENTS

Amazing Mail 6

Amiga User Groups 12

Index of Advertisers 96

Public Domain Software Catalog 109

"I've written the best and fastest backup program on the market."

My name is Walt Soden. I've been a programmer for thirty years, and I know how important it is to back up your hard disk. But when I looked for a good backup program for my Amiga, I found they took too much of my time managing the backup disks. I knew there had to be a better way--so I spent a year writing and perfecting what I sincerely believe to be the best backup software available.

EZ-Backup

Does what Quarterback can't.

EZ-Backup does ALL the work. EZ-Backup knows which files to back up, how many versions to save, which to erase and where they are in the backup set. So you only have to keep one set of backup disks, period. Your files will always be there, safe and sound--in the standard AmigaDOS format.

A special offer.

Is it the lowest price and the best value in backup software? You be the judge. **EZ-Backup** comes with free phone support. If you have any questions, just pick up the phone and you can talk to me personally. I'll send you a working demonstration copy of EZ-Backup (limited only in the number of files it can back up) for only $5.00. If you like it, then take advantage of the special discount offer explained on the demonstration disk or you can buy the full version from your local Amiga dealer. If you don't think it's the best, most convenient backup software you've ever tried, send back the disk and I'll refund your five bucks.

EZ-Soft, 10668 Ellen Street, El Monte, CA 91731, (818) 448-0779.

Quarterback is a trademark of Central Coast Software.

EZ SOFT

Circle 119 on Reader Service card.

Amazing COMPUTING

For The Commodore AMIGA™

ADMINISTRATION

Publisher: Joyce Hicks
Assistant Publisher: Robert J. Hicks
Circulation Manager: Doris Gamble
Asst. Circulation: Traci Desmarais
Asst. Circulation: Donna Viveiros
Corporate Trainer: Virginia Terry Hicks
Traffic Manager: Robert Gamble
International Coordinator: Marie A. Raymond
Marketing Manager: Ernest P. Viveiros Sr.
Administrative Support: Aurore R. Trepanier

EDITORIAL

Managing Editor: Don Hicks
Associate Editor: Elizabeth Fedorzyn
Hardware Editor: Ernest P. Viveiros Sr.
Technical Editor: J. Michael Morrison
Music & Sound Editor: Richard Rae
Video Editor: Barry S. Solomon
Copy Editor: Aimée B. Abren
Copy Editor: Derek J. Perry
Copy Editor: Matt Rita
Art Director: William Fries
Photographer: Paul Michael
Illustrator: Brian Fox
Production Manager: Donna M. Garant

ADVERTISING SALES

Advertising: Jannine Irizarry
Special Assignment: Barry Solomon
Marketing Assistant: Melissa J. Bernier

1-508-678-4200
FAX 1-508-675-6002

SPECIAL THANKS TO:
Buddy Terrell & Byrd Press
Bob at Riverside Art, Ltd.
Swansea One Hour Photo
Pride Offset, Warwick, RI
Mach 1 Photo

Amazing Computing™ (**ISSN 0886-9480**) is published monthly by PiM Publications, Inc., Currant Road, P.O. Box 869, Fall River, MA 02722-0869.

Subscriptions in the U.S., 12 issues for **$24.00**; in Canada & Mexico surface, **$36.00**; foreign surface for **$44.00**.

Second-Class Postage paid at Fall River, MA 02722 and additional mailing offices.

POSTMASTER: Send address changes to PiM Publications Inc., P.O. Box 869, Fall River, MA 02722-0869. Printed in the U.S.A.

Copyright©Nov. 1988 by PiM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request. PiM Publications, Inc. maintains the right to refuse any advertising.

Pim Publications Inc. is not obligated to return unsolicited materials. All requested returns must be received with a Self Addressed Stamped Mailer.

Send article submissions in both manuscript and disk format with your name, address, telephone, and Social Security Number on each to the Associate Editor. Requests for Author's Guides should be directed to the address listed above.

AMIGA™ is a registered trademark of Commodore-Amiga, Inc.

Amazing Dealers

The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™. They carry Amazing Computing™, your resource for information on the Amiga™. If you are not an Amazing Dealer, but would like to become one, call PiM Publications, Inc.:

1-508-678-4200

Alaska	
That Computer Place	Anchorage
Juneau Electronics	Juneau
Alabama	
Abax Data Systems	Huntsville
Alabama Book Store	Tuscaloosa
Madison Book & Computer	Madison
Mel's Photo & Computer Shop	Montgomery
The Computer Nut	Birmingham
The Computer Image	Birmingham
Universal Computer System	Mobile
Arkansas	
Commodore Connection	Sherwood
Microtronix	Fort Smith
The Micro Shop	Little Rock
Arizona	
Books Etc.	Tempe
Books Brothers Ltd.	Tucson
Computer Libray, Inc.	Tempe
Copperstate Business Systems	Phoenix
North American Digital	Tucson]
Try Tech Inc.	Pheonix
California	
3 E Software And Systems	Hayword
American Mags & Books Ltd.	Englewood
Balboa Office Supply	San Diego
Bits & Bytes	Fairfield
Bits n Pieces Comp Prod.	Anaheim
Bookland	Marysville
Brown Knows Computing	Redlands
CAL-Tech Bookstore	Pasadena
Candy Computer	Elk Grove
Centerfold News	Los Angeles
Central Park Bookstore	San Mateo
Century Computer Systems	La Habra
Circus of Books	Los Angeles
Circus of Books	Los Angeles
Commodore Land	Alhambra
Commotech	National City
Computer Attic	Palo Alto
Computer Literacy	Santa Clara
Computer Literacy Books	Sunnyvale
Computer Nook, Inc.	San Bernadino
Computer Outlet	Chula Vista
Computer Outlet	San Diego
Computer Outlet	San Marco
Computer Spectrum	Burlingame
Computer Time	Citrus Hts
Coronet Newsstand	Oceanside
Covina Bookstore	Covina
Dataphile	Watsonville
Delaurer's News Agency	Oakland
Discount Computer Station Inc.	Long Beach
Empire Computers	Santa Rosa
Encinitas News	Encinitas
ECX Computer Co.	Walnut Creek
Fisher Computers + Software	Rancho Cucamonga
Harding Way News	Stockton
Heinrichson, Schneider & Young, Inc.	Berkeley
Home Computing Center	Concord
Home Computing Center	Mill Valley
Home Computing Center	San Leandro
HT Electronics	Sunnyvale
International Telecine	Santa Ana
K J Computers	Granada Hills
Little Professor Books	Placentia
LTC Computers	Santa Ana
Lyon Enterprises	Berkeley
Micro Flight	San Jacinto
NewsBeat	Davis
Northridge Newsstand	Northridge
Op Amp Tech Books	Los Angeles
Paras Newsstand	San Diego
Put's Electronics	North Highlands
R&R Software	Chula Vista
Ridgecrest Computer Ctr. Inc.	Ridgecrest
Sawyer's News Inc.	Santa Rosa
Software First	Santa Rosa
Software Cellar	Anahiem
Software Central	Pasadena
Software Plus	Citrus Hieghts
Software Solutions Plus	El Cerito
S.O.S. Computers	Los Angeles
Stacy's #5	Modesto
Technitronics Service	South Pasadena
The Booksmith	San Francisco
Third Avenue News	Chula Vista
Tower Books	Chico
Tower Books	Citrus Hts.
Tower Books	Concord
Tower Books	Mountain View
Tower Books	Sacremento
Tower Books	San Mateo
Universal News	Hollywood
WCI Computer Central	Eureka
Warner Engineering	San Diego
World Book & News	Hoolywood
Colorado	
Aurora Newsstand	Aurora
Citadel Computers	Colorado Springs
Computer Discount	Denver
Computer Repeats	Boulder
Computer Room	Aurora
Micro Computer World	Fort Collins
Micro World	Lakewood
Sun Country Computers	Colorado Springs
Connecticut	
Aetna Life Club Store	Hartford
Amigo Business Computer	Norwalk
Computer & Software Connections	Plantsville
Derric Electronics	Hamden
Friendly Computers	Plantsville
Inlotronics, Inc.	Woodbury
Mnemonics Inc.	Norwalk
One Byte	Quaker Hill
Personal Computer Center	Norwich
Software City	Orange
Software Kingdom	East Hartford
Tricom Computers	North Windham
District of Columbia	
Prograna	Washington
Delaware	
Castle Video	Newark
Delaware Electronics	New Castle
Newark Newsstand	Newark
Florida	
A A Computers	Jacksonville
Ami Comp Computer Center	Oviedo
Chuck's Newsstand	Vera Beach
City Newsstand	Melbourne
Commodore Country Inc.	Pinellas Park
Computer Bar	Pennsacola
Computer Base	Brandon
Computers Etc.	Sarasota
Computer Image	Miami
Computer Terminal Inc.	Sebring
Computers + Video Creations	Palm Beach
Computers 4 Rent	W. Palm Beach
Computers Plus	S. Daytona
Creative Equipment	Miami
Education Computers Etc.	Talahasse
Florida Book Store	Gainesville
Gulf Coast Computer Exchange	PanamaCity
Joe's News, Inc.	Miami
Micro Computer Center	Largo
New Age Electronics	Clearwater
New Age Electronics	St. Petersburg
PC Collection	Coral Gabels
Program Store	Clearwater
Prospect News & Books	Ft. Lauderdale
RGB Computers and Video Creations	Palm Beach Gardens
Sunny's At Sunrise, Inc.	Sunrise
Computer Room	Ft. Lauderdale
The Open Door	Merritt Island
Georgia	
BFJ Enterprises	Roswell
Cardinal Computers	Dalton
Personal Computer Exchange	Lilburn
Software South Inc.	Savannah
The 64 Store	Atlanta
Hawaii	
Astro Computers & Electronics	Aiea
Computronics	Honolulu
Data-One Computer Center	Honolulu
J & C Repair	Aiea
Idaho	
A.B.I. Computer & Video	Nampa
Illinois	
Audio Visual Req.	Chicago
Comp—Kingdom	Lisle
Computer Software Connection	Moline
DigitalWorld	Addison
DOS Computer Center	Niles
Keep In Pace Computers	Champaign
Kroch's & Brentano's	Chicago
Kroch's & Brentano's	Schaumburg
Kroch's & Brentano's	Orland Park
Kroch's & Brentano's	Auroa
M2J2 Inc.	Champaign
Micro Age	Oaklawn
Micro Ed Enterprises Inc.	Chicago
Micro Tek	Kankakee
Micro Tech Soultions	Aurora
Ring Software	Geneva
Software City	Belville
Software Plus	Wheeling
Software Plus Chicago	Chicago
Software Plus West	Hanover Park
The Memory Expansion	Bloomington
Unique Computer	Sterling
Warren Radio/FRANCO	Peoria
Indiana	
Bookland	Indianoplis
Burkat Computer Ctr.	South Bend
Computer Corner	Ft. Wayne
Computer Corner	Marion
Computer People Inc.	Michigan City
CPU Inc.	Indianapolis
Indiana News	Indianapolis
Micro Computer Centers Inc.	Indianapolis
Reese Equipment Co.	Plymouth
The Book Corner	Bloomington
The Computer Room of Merrivile	Merrville
Vons Computers	West Lafayette
Iowa	
Gronert Computers	Des Moine
Soft Search	Ames
Kansas	
Arc Electronic Distributors	Wichita
Midwest Computers	Manhattan
Software City	Overland Park
Kentucky	
Computer Dimensions Inc.	Lexington
Computer Mania Inc.	Florence
Hawley Cooke	Louisville
Reality World Computers	Lexington
The Computer Lab	Louisville
Louisiana	
City Newstand	Baton Rouge
Computer Time	Shreveport
Lakeside News	Metairie
Modern Business Machines	Metairie
Software Center International	Metairie
Software Source	Lafayette
The Book Shelf	Sidell
The Computer Clinic	Lafayette
Maine	
Computer Barn	Damariscotta
Hands On Computing	Westbrook
Skylight Software	Belfast
Maryland	
Buried Treasure	Rockville
Cal Com Inc.	Silver Spring
Greetings and Readings	Towson
Knowledgeware Systems	Kensington
Micro Computer Center	Baltimore
New Age Computers	College Park
Professional Micro Services	Baltimore
Software Advantage	Rockville
Waldorf Computer	Waldorf
Massachusetts	
Computer Works	Middletown
Computer Zone	North Attleboro
HCS Computer Center	Marshfield
LCA Video & Computer Ctr.	Norwood
Memory Location	Wellsley
OmniTek Computers	Tewksbury
Pioneer Valley Data Equipment	Amherst
Soft Designs	Watertown
Softpro	Burlington
The Bit Bucket	West Newton
The Software Haus	Quincy
Tycom Inc.	Pittsfield
Michigan	
CompuSoft	Kalamazoo
Computer Emporium	Lansing
Direct Access	Lansing
Central Park Computers	Okemos
Global Computer Center	Dearborn
I/O Softwware	Flint
Michigan News Agency	Kalamazoo
Michigan Software	Novi
Micro World	Livonia
Micro World	Milford
Micro World	Roseville
Micro World	Southfield
Micro World	Southgate
Micro World	Sterling Heights
Microworld Computer Center	Dearborn
Midland Computer Shop	Midland
Norman Business Control Sys.	Wyoming
Readmore Newscenter	Lansing
Rochester Book Center	Rochester Hills
Roseville Computer Store	Warren
Slip Disk	Madison Heights
Software City	Grand Rapids
Specialists Computer	Livonia
State Street Computers	Ann Arbor
T.J. Enterprises	Clinton
The Edge Connector	Clio
Ye Olde Computer Shoppe	Ypsilanti
Minnesota	
Computer Excercise World	Minneapolis
Computer Stuff	Minneapolis
Computers, etc.	Eagan
JMH Software	Maple Grove
Shinder's	St. Paul
Shinder's	Maplewood
Shinder's	Burnsville
Shinder's	Minnetonka
Shinder's	Edina
Shinder's	Crystal
Shinder's Read More	Minneapolis
Specialists IN	Hopkins
Valiant Inc.	Stillwater
Mississippi	
Computer Connection	South Haven
National Computer Center	Ocean Springs
Missouri	
Brands Mart Computers	Kansas City
Carthage Electronics	Carthadge
Data Byte Computer Co.	St. Louis
Software City	Independence
Systems Plus	Overland Park
Systems Plus of St. Louis	St. Louis
Nebraska	
Double E Electronics	Omaha
My Com Inc.	Plattsmouth
Nevada	
Computer House	Sparks
Computer World	Las Vegas
Tower Records	Las Vegas
New Hampshire	
Dartmouth Bookstore	Hanover
Diversified Computers	Keene
System Eyes	Merrimack
New Jersey	
Act One	Vineland
Business System Professional	Scotch Plains
Family Computer Centres	South Orange
Morris County Stationers	Chester
Software Station	Rockaway
The Program Store	Eaton Town
The Program Store	Wayne
Unlimited Software Inc.	Rutherford
New Mexico	
New Horizon Computer System	Alamogordo
Page One Newstand	Albuquerque
Road Runner	Espanola
New York	
Amicus- Hudson Valley	Wappingers Falls
Amigo Business Computer	East North Port
Byte Shop	Merrick
Compurite Center	White Plains
Computer Outlet	Jamestown
Computers Etc.	Syracuse
Computerware	East Meadow
Contemporary Computing	Latham
Creative Computers	Johnson City
Cromland Inc.	Brooklyn
Global Software	New Hyde Park
Island Software Computer Services	Coran
Leeder's Video	Schenectady
Leigh's Computers	New York
MCP Associates	Astoria
MicroWorks	Kenmore
Mnemonics	North Port
Readers Market	Tonawanda
Software & Such	Scotia
Software City	Forest Hills
Software Link	White Plains
Software Supermarket	Tonawanda
Star Tech Systems	Massena
Synfaire Trading	New York
Sysput Computers	Brooklyn
Tadiran	New York
The Computer Cellar	Albany
Unicorn Electronics	Johnson City
Video Computer Bin	Brooklyn
Video Computer Center	Rome
Village Computer Inc.	New York
World Computers Inc.	Hicksville
North Carolina	
Digitz	Raleigh
K&S Newsstand	Clemmons
K&S Newsstand	Kernersville
K&S Newsstand	Winston-Salem
News Center	Cary
Parker News Inc.	High Point
Software City	Raleigh
Software City	Charlotte
Triad Computers	Winston-Salem
North Dakota	
Computer Associates	Fargo
G. F. Computers	Grand Forks
Software House	Fargo
The Computer Store	Minot
Ohio	
B5 Software	Columbus
Computer Network	Newark
Computer Site	Cleveland
Eddy's Cigar Store	Elyra
Fairborn Home Computer	Dayton
Lakewood International News	Lakewood
Lakes Consumer Electronics	Akron
Leo's Book & Wine Shop	Toledo
Microtech Computer Center Inc.	Toledo
Microware Magic	Fairfield
News Shop	Kent
North Coast Programming	Willoughby
Paper & Ink	Tiffin
Softwaire Centre International	North Olmsted
Software and More	Cincinnati
Software Connection	Mayfield Hts.
Somers Photo & Comp.	Cleveland Heights
Specialist Computer System	Toledo
Oklahoma	
Little Proffessor Books	Stillwater
Merit Computers	Oaklahoma City
Micro Link Computer Store	Tulsa
P C Tech	Tulsa
Second Hand Software	Oklahoma City
Video-Comp Inc.	Lawton
Oregon	
Andromeda	LaGrande
Clackamas Computers	Beaverton
Clackamas Computers	Clackamas
Comm-Shack	Salem
Computron Business Systems	Portland
PC Express	Eugene
Powell's Tech	Portland
Rich Cigar Store	Portland
Sixth & Washington News	Portland
Software Pipline	Beaverton
Tech Books	Portland
Pennsylvania	
Alpha Omega EBE	Warren
ATD Software	Sayre
Basic Computer Systems	Erie
Basic Computer Systems	Hermitage
Circle Computer	Ephrata
Computer Dimensions	Altoona
Computer Wurx	Stroudsburg
Computers and Games	Reading
Downington Computer Center	Downington
Elmo's Computers	Brownsville
Floppy House Software	Palmyra
Gene's Books	King of Prussia
Lauer-Todd Computer Solutions	Newtown
Micro Master Inc.	Monaca
North Central Digital Systems	Danville
Pittsburgh Computer Store	Pittsburgh
Software Hut Inc.	Philadelphia
Some Hole In The Wall	Philadelphia
Tasco Computer Center	Lancaster
Triangle Computers	Indiana
Walden Hill Images	Berlin
Rhode Island	
Micro Limits	Smithfield
The Computer Place	Woonsocket
Software Connection	Warwick
South Carolina	
Avcom International Inc.	West Columbia
Watson's Jewlery	Chesterfield
Software Haus	Charleston Hts
Tennessee	
ComputerLab of Memphis	Memphis
Computer Hardware Services	Nashville
Data International	Hickson
The 64 Store	Bartlett
Texas	
A&E Computers	Grove
All Systems Go Software	San Antonio
Amazing Computer Systems	Carrollton
Bookstop	Houston
Bookstop	San Antonio
Byte Shop	Austin
C.E.M. Corporation	Brazoria
Computer Age	Houston
Computer Magic	Austin
Computer Skills	Eules
FNS Computer Services	Texarkana
IMaxwell Book Inc.	Desota
Metropolitan Computer	Richardson
Software Library	Witchita Falls
Software Terminal	Fort Worth
The Computer Experience	San Antonio
The Computer Stop!	Amarillo
The Federated Group	Mosquite
Utah	
Magazines and Gifts	Salt Lake City
People's Computers	Provo
Vermont	
Ray Supply Inc.	South Burlington
Virginia	
Audio Fidelity Corp	Richmond
Family Computer Center	Fairfax
Family Video & Computer Center	Newport News
I/O Computer	Norfolk
Software City	Richmond
Tract Software	Prince William
Video Computer Retailers	Chesapeak
Virginia Micro Systems	Woodbridge
Washington	
Amazing Electronics	Spokane
A.P.P.L.E. CO-OP	Renton
Apple Valley Computers	E. Wenatchee
Bits, Bytes and Nibbles	Spokane
Bull Dog News	Seattle
Byte World	Kent
Computer Quest	Edmonds
Computers + Inc.	Tacoma
Family Computers	Everett
Micro Age Computer Store	Bellevue
Nybbles & Bytes	Tacoma
Omni International	Seattle
Port Book and News	Port Angeles
Real Time Systems	Seattle
Take A Byte	Kennewick
Tech Star	Kent
The Computer Shoppe	Vancouver
Tower Books	Bellevue
Y Not Magazines and Boks	Federal Way
Wisconsin	
20th Century Books	Madison
Computer Software Center	Milwaukee
Fox Valley Personal Computer	Oshkosh
Madison Programmer's Guide	Madison
Mayfair Computers	Milwaukee
Mom's Computers	Stevens Point
River City Software	La Crosse
Software City	Racine
TMW Software Inc.	Wausau
Wyoming	
Byte Emporium	Laramie
Computer Works	Laramie
Outside of the United States	
Canada	
Compute or Play	Alberta
Computerworks	Alberta
Computer Shop	Alberta
Data Pro	Alberta
Computer Shop of Calgary	Alberta
The Software House	Alberta
Hindsin Computing	Alberta
Red Deer Micro Systems	Alberta
Memory Land	Alberta
Compu Care	British Columbia
Computer King	British Columbia
Conti	British Columbia
Moebius Computer Systems	British Columbia
Nutec	British Columbia
Pacific Business Technologies	British Columbia
Sprita	British Columbia
Strider	British Columbia
Super Software	British Columbia
Canadian Computer Era	Manitoba
Micromart	Manitoba
ComputerWorld	Manitoba
J.W. Allen	Manitoba
Kobetak	Manitoba
Minicomp	Manitoba
Arkon	Ontario
Chatam Computer	Ontario
Computerland	Ontario
Comspec	Ontario
Electronics 2001	Ontario
Hastings Data	Ontario
The Computer & You	Ontario
Thornhill	Ontario
Software Co.	Quebec
Deltronie Solare	Quebec
Merodata	Quebec
Info Mereier	Quebec
Laval Micro	Quebec
Vision Tronique	Quebec
Distribution Visipro	Quebec
Distributed in Canada by	
Ingram Software Distributors	Concord, Ontario
Compulit Distributors	British Columbia
Phase 4 Dist. Inc.	Calgary, Alberta
Argentina	
Logic Computer	Buenos Aires
Australia	
Computermart	Leederville,WA
Diskworks	Greenwich, NSW
John Verhoeven	Cloverdale, WA
Maxwell Office Equipment	Abbotsford, Melburn
Run Soft	Melbourne
Saudia Arabia	
Abdulla Fouad & Sons	Daman
Scotland	
The Amiga Center of Scotland	Edinborough
Sweden	
Nicholas Peerson	Lund
Procomp	

Amazing Computing™ is also available in most B. Dalton Booksellers, B. Dalton Software Stores, Crown Books, Software Etc., selected WaldenBooks Stores, and Walden's Software Store locations.

Dear AC:

The articles in Vol. 4 No.4 concerning Amiga hard drives was excellent material and helped to make this your best issue yet. Please publish more of these types of articles in the future.

The information presented by Boulle, Twardy, and Morgan will undoubtedly whet the storage appetites of Amiga users everywhere. A great number of Amiga users badly want a hard drive for their favorite machine but cannot spring for the high price of a store bought unit. Others of us refuse to settle for an affordable but paltry 20 megabyte drive and want/need a higher capacity drive that is financially attainable.

Information such as you have presented in your last issue will be of great help to Amiga users who'd like to "roll their own" hard drive system but need some help to do it. As good as these articles were, there is a gigantic fly in the ointment! Yes. Several suppliers of HD system parts and prices are mentioned, but there aren't any addresses or phone numbers that tell us WHERE to locate these vital parts! Arrgghhhhh!!!!

So far, I have combed Computer Shopper for an OMTI 3527 RLL-SCSI adapter/controller with no luck at all...plenty of slow Adaptec 4070's for sale but no OMTI's Also, a price of $149 was quoted for the Supra host adapter. The lowest price that I have seen on this is $185. Please let me know where I can save a few bucks and get it for $149. I did manage to find Wetex in the CS, so I can get the case/PS but who and where are DigiKey and L.S.I. Marketing?

Thanks in advance for any additional info that you can send my way. I'm sure that lots of AC readers would appreciate such additional information as well.

Sincerely,
Edward E. Brown
Vancouver, WA

-We contacted Jon Boulle, and he said he buys OMTI's from C-Ltd for about $185. He suggests that you may want to purchase an imbedded SCSI device now that the prices are down.—ED

Dear AC:

I am writing in regard to an article in the March 1989 issue entitled "Fractal Fundamentals", by Paul Castonguay. I found the article easy to understand and I had fun with the program and generating my own fractal designs. I have found similar articles on computer-generated fractal patterns in the *Scientific American* "Computer Recreations" column frustrating because of their refusal to explain how to generate the fractal patterns.

The article in AC let me program my computer to perform the surprisingly simple operations. The article really bridged a gap between the difficult to apply discussions and the interested but easy hacker who wants to see the patterns on his own monitor.

The article in the March issue was billed as "Part I," but I have yet to see Part II. I hope to see the next installment soon.

Sincerely,
Daniel Tormey
Cambridge, MA

-Fractals Part II was printed in AC V4.7 on page 95, and Fractals Part III will appear soon in a future issue. Paul is doing a great job, and we currently plan to print up to Fractals Part VI, unless you want us to print more. Let us know.—ED

Dear AC:

I thoroughly enjoyed the article by James Bayless in V4.2, "A Common User Interface for the Amiga". His plea for a polished, consistent presentation of programs must be heeded if the Amiga is to enjoy continued success in a broad market.

For every hacker and hardware-head who bought this machine because of its technical excellence (like myself) there are a thousand other people who want a computer that will simply allow them to get their work done quickly and smoothly. These folks don't care about advanced hardware inside their machine, the amazing algorithm you developed for text-searching, or how you cleverly circumvented the operating system to speed up file access. What they do care about is whether or not a program is easy to master, easy to use, and gets the job done.

If they are confronted with cluttered, garnish screens, a bewildering array of gadgets whose functions aren't immediately obvious, or opaque terminology, they will come away confused, frustrated, and angry. Who can blame them? For the vast majority of people, a computer is (and should be) a tool, not an end in itself. No one needs to know anything about chemistry, optics, or electronics to use a camera. They shouldn't have to know anything about blitters, bit-maps, or Bressenham algorithms to write a letter or draw a picture using their computer.

The user interface should be developed first, before any other code is written. It should not be left until last, and used to glue together all of the routines that comprise the program. Fit your routines to the interface so that your program has coherence and consistency. Go ahead and make your programs powerful, but make access to that power simple. Above all, strive to make your programs beautiful and a joy to use, rather than a chore. Spend a hundred hours getting your presentation "just so". Then spend another hundred hours polishing it.

Sincerely,
Paul M. Carlisle
Royal Oak, MI

-Commodore agrees with this philosophy and is including a programmers library in the V1.4 release. The ASL.library will include items commonly needed by programmers, such as a file requestor. This will make the user feel comfortable moving from one application to another. —ED

Dear AC:

I enjoyed your recent construction article for an Amiga Stereo Sound Digitizer. This was an excellent article, especially since a PC board layout was included.

However, may I make a suggestion? Next time, please make the PC layout

Got The Picture...Get The Works!

PLATINUM EDITION™

Picture this: all the productivity applications you need in one easy to use "Starter Kit". Give your Amiga power times five. Give it **The Works! Platinum Edition.**

✓**WORD PROCESSOR** • The Works! Platinum Edition word processor is powerpacked with features that help the serious writer excel, in an easy-to-use environment that makes beginners instantly productive. The 104,000+ word dictionary with Scientific and Technical supplements keeps your spelling picture-perfect. The 470,000+ word thesaurus with definitions keeps your word-images precise. Mail Merge eliminates repetitive typing. All this, the ability to print IFF graphics, and more! It may be all the word processor you'll ever need.

✓**SPREADSHEET** • The Works! Platinum Edition spreadsheet is lightning fast — in fact, the fastest Amiga spreadsheet. And it supports the 68881 co-processor for even more blinding speed. The more than 40 built-in functions do sophisticated calculations. Picture your numbers in any of eight graph types and displayed in eight vibrant colors. Its complete Macro language automates complex operations.

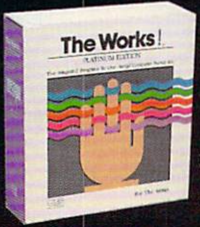
✓**TELECOMMUNICATIONS** • The Works! Platinum Edition is a sophisticated telecommunications program. It is a special part of the outstanding whole. Its multiple terminal emulations and *ten* transfer protocols are the hallmark of flexibility. The script language offers unattended operation and the user defined Macro-Keys reduce complex commands to a single keystroke. Ask about Sadie.

✓**DATABASE** • The Works! Platinum Edition is a flat file manager with power. The extensive mathematical functions make reports much more than a simple list of data.

✓**SIDEWAYS** • The fifth power module in The Works! Platinum Edition, stands your printouts on end. Print ASCII text files and IFF Graphics rotated 90-degrees.

The Works! Platinum Edition is true *integration* from the Micro-Systems Software Development Team; pioneers in Amiga productivity products. Experience full Clipboard compatibility, a common interface, and one user friendly manual. You owe yourself the Platinum experience! Check out **The Works! Platinum Edition** at a dealer near you.

Committed to excellence since 1978



MSS
Micro-Systems Software

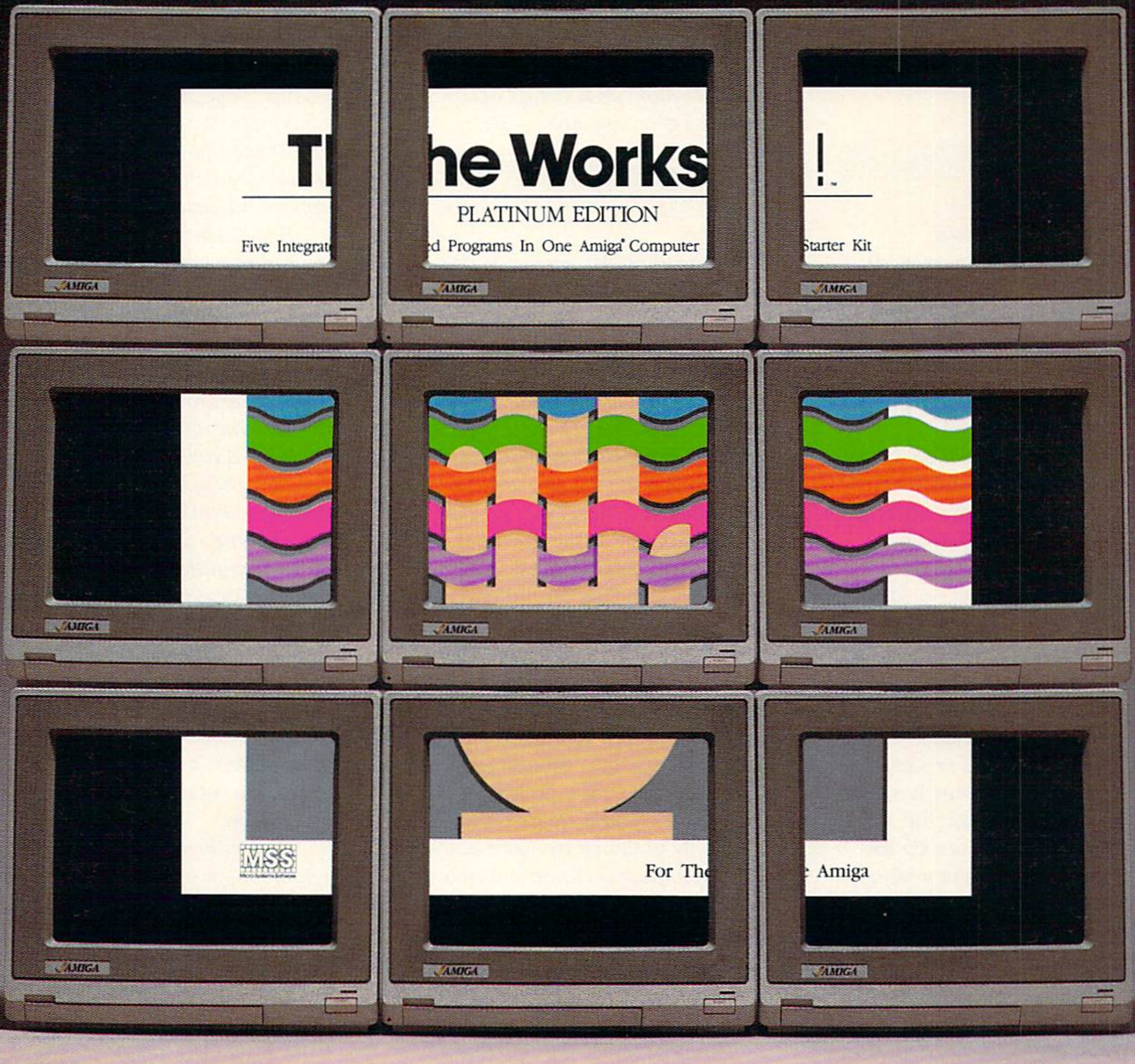
12798 Forest Hill Blvd., Suite 202
West Palm Beach, Florida 33414
407-790-0770 Fax 407-790-1341

Dealers and Distributors Call 1-800-327-8724

See your local dealer for a demonstration.

We use KAO Disks.

The Works! Platinum Edition is a trademark of Micro-Systems Software, Inc. All brand and product names are trademarks of registered trademarks of their respective companies.



Circle 126 on Reader Service card.

actual size, and print the layout right-side up (foil side toward the reader). There was no easy way for your technically inclined readers to easily make a PC board from the illustration you provided. It would have involved making a PMT of the page (after properly scaling the layout) and flipping it over to make a board. Also, when you print construction articles, please try to get PC layouts for the projects. They make construction much easier.

Your magazine is a great informational source for the Amiga. Thanks to you, I have an audio filter switch installed in my A1000, and will be able to manipulate my dog's bark beyond recognition. Keep up the good work!

Sincerely,
Paul Combs
Dayton, OH

-Thank you for these tips. We received other letters recommending the same ideas. We are one of the few Amiga magazines that still does hardware projects and we plan to continue. We will use these suggestions in future hardware articles.—ED

Dear AC:

I have had a CMI Processor Accelerator with a 68881 running at 20 MHz for about 6 months now and it's great. I just read the review on it and would like to make a few comments. It is true that the 10-30% speed increase is somewhat minimal, but this is exactly as advertised.

Secondly, Rich Grace seemed a little unsure of how the coprocessor interface works with the 68881 hardware floating point chip. The 68000 CPU does not have a coprocessor interface (as does the 68020, 30 and 40) so the 68000 must use the 68881 as a peripheral which is much slower than using it as a coprocessor, but still much faster than doing your floating point math without the 68881.

Programs do not have to be especially tailored to the CMI Processor Accelerator. New programs only have to be compiled in a way as to use the floating point library that comes on the Workbench disk on 1.3. This new library will take advantage of a 68881 if there is one. All new programs should (hopefully) be compiled this way so that users can see a significant speed increase in any program using floating point. Lattice 5.0 has an option to compile a program in this way, and I've heard that Aztec C also has this option in its newest release.

Using this method to compile is not specifically for the CMI board, but will speed up floating point programs regardless of what brand of board you have that has a 68881 on it. I have seen some of my benchmarks show as much as 70 times increase in speed, but a more realistic speed increase is three to six times faster for a regular program which does a lot of floating point.

Overall I liked the review and I certainly like the CMI Processor Accelerator. For anyone interested in a program which will show the floating point speed increase, I have written a program called galaxy collision simulator. It is on the Lattice BBS and Compuserve (under the AMIGATECH section).

Steve E. Riley
Shingle Springs, CA

-More programs are being written to take advantage of both math coprocessor chips and accelerator cards. Sculpt 4D by Byte by Byte is a good example.—ED

Dear AC:

After reading Barney Schwartz's review of "PageStream," I came away with mixed emotions. On one hand, a review by the person who wrote the manual for the program is sure to be informative. He describes all the wonderful features very well. On the other hand, a person with such close ties to the manufacturer can be very easily biased in his opinions, as is the case when he implies that the limited number of proprietary fonts supplied with the program are sufficient for professional use.

He and SoftLogic have made a big mistake. In this case, ignoring the thousands of Postscript fonts available from respected type foundries such as Typeface Corporation, Monotype, Compugraphic, Letraset, Adobe, Bitstream, etc. These are all typographic quality fonts designed by some of the world's best artists, refined and tested over many years of actual use by professional graphic designers and typographers. To ignore this mass of quality fonts that are available yesterday and to substitute a measly ten fonts of dubious quality is pointless. The "promise" of more fonts "real soon now" is not very comforting.

In the graphics arts business the quality of the final output is all that matters. When you spend thousands of dollars for printing that is vital to your business, you want the best type possible. As a professional graphic artist I cannot expect my clients to settle for second class typography just because my page layout doesn't allow the use of a wide variety of quality fonts.

Until Amiga software developers and Commodore themselves take full advantage of Postscript fonts, Amiga desktop publishing programs will be just second rate.

Sincerely,
Andre Page
Sepulveda, CA

-Here is a response from Soft Logik:

In response to the letter from Andre Page we would like to clear up a simple misunderstanding concerning our product. PageStream is FULLY PostScript compatible. We do not ignore or hide from PostScript or its fonts. We in fact strongly recommend its use in professional settings as Andre cited.

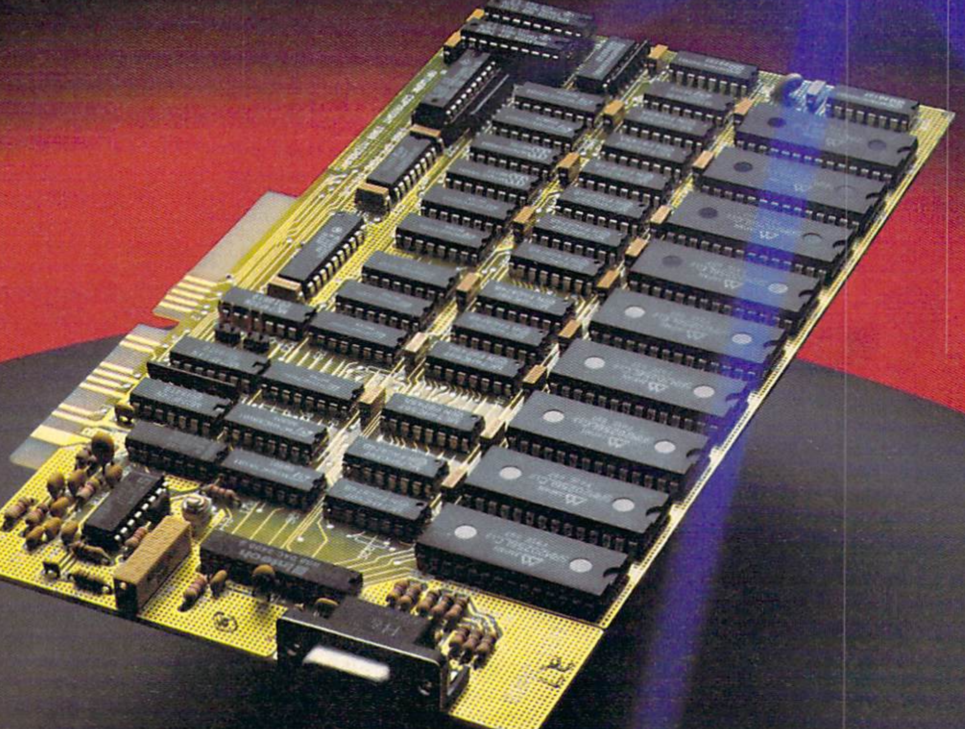
It seems that the misunderstanding stems from a paragraph in Barney Schwartz's article. In the article he states that "...you are limited to PageStream fonts within your document." This is true in that PageStream must have a screen font for every font that it prints out, as is true with any DTP package, on any machine. But that does not mean you are limited to Soft-Logik's outline fonts when printing to PostScript printers.

Currently we offer screen fonts for the 35 "standard" fonts that come in most PostScript printers. Additionally we offer many more downloadable PostScript fonts. But even with this wide selection we do not have screen fonts for the several thousand typefaces available to PostScript imaging devices. In fact, other than for PC's and Mac's, there exist no standard screen fonts at all. And that is the core of the problem.

Currently none of the major type vendors support the Amiga like they do other computers. Soft-Logik for one would be over joyed if they would. On a practical side, if Soft-Logik were to try to

AGA 2000

The flickerFixer™



UNLOCK THE GRAPHICS POWER OF YOUR AMIGA 2000!

flickerFixer is an advanced graphics adaptor that eliminates your Amiga 2000's interlace flicker and visible scan lines. The result: superior quality color or monochrome graphics and text — for such demanding applications as CAD/CAM, Desktop Presentation, Graphic Design, Animation, 3D Modeling, Video, and Word Processing.

flickerFixer upgrades the Amiga 2000 with a flicker free 4096 color palette, has an overscan mode that features a screen size of 704 x 470 pixels and drives most of the popular PC Multiscan and VGA monitors, including the NEC Multisync and Mitsubishi XC1429C.

Accolades include: ***Best of 1988 Award,*** Commodore Magazine (12/88); ***#1 Reader's Choice Award***, AMIGAWORLD (12/88); ***"The display is fantastic . . . It is the best display we have ever seen on any computer system."*** Amiga GURU (5/88)

flickerFixer fits into the Amiga video slot, is fully compatible with all software, and does not modify the standard Amiga video signals. For more information or to order, call MicroWay Sales at (508) 746-7341 or your Amiga Dealer. Priced at $595, ***flickerFixer*** is made in the USA and is FCC Class B approved.

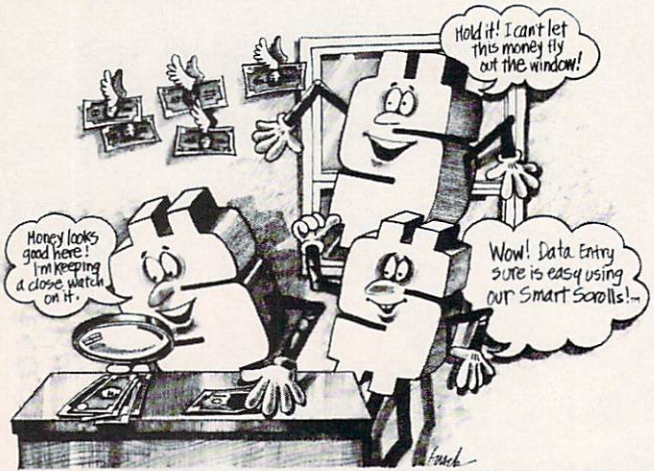
MicroWay . . . Respected throughout the industry for high quality engineering, service and technical support.

MicroWay

World Leader in PC Numerics

P.O. Box 79. Kingston, MA 02364 USA (508) 746-7341
32 High St., Kingston-Upon-Thames, UK, 01-541-5466
USA FAX 617-934-2414 Australia 02-439-8400

flickerFixer and MicroWay are trademarks of MicroWay, Inc. Amiga is a registered trademark of Commodore. Multisync is a registered trademark of NEC.
Circle 114 on Reader Service card.


Hold it! I can't let this money fly out the window!
Money looks good here! I'm keeping a close watch on it.
Wow! Data Entry sure is easy using our Smart Scrolls!

Meet a team of the friendliest financial organizers you'll ever run across.

When you want to manage your personal finances, Money Mentor goes a step beyond.

Plug Money Mentor into your Amiga and a virtual teamwork effort takes place in watching over every aspect of your personal finances.

The new "C" version of Money Mentor is the friendliest financial organizer obtainable today!

Now you can experience super-speed data entry, dazzling graphic output and an extremely friendly attitude!

Smart Scrolls for speed.

Money Mentor has a truly unique system called *Smart Scrolls*, that handles a diversity of otherwise tedious data entry functions and clips along saving you up to 70% of your typing time. It's a *smart* addition to Money Mentor, that's why we call it *Smart Scrolls*.

Money Mentor Features:

- Net Worth Statement
- 200 Budget Categories
- 30 Integrated Accounts such as Checking, Cash, Savings and Credit Cards
- Elaborate Search Routine allows editing of transactions according to your specific guidelines
- Automatic Check Printing
- Automatic Account Balancing
- Color Graphic Reports illustrating *actual vs. budgeted* amounts
- Over 50 Reports to choose from!

What they're saying about us!

"Money Mentor has to be the nicest look and feel of any money manager package for home use that I have ever seen." — **Amiga Sentry**

"Money Mentor is an excellent product" — **Amazing Computing**

Money Mentor is for everyone!

It does more than just keep your checkbook balanced. Money Mentor helps you manage your personal finances which is important to any family or individual.

With Money Mentor, you can be looking better financially.

Order Money Mentor today.

Money Mentor sells for only $95.95!

VISA MasterCard

SEDONA SOFTWARE

SEDONA SOFTWARE/11828 RANCHO BERNARDO RD., SUITE 128-20/SAN DIEGO, CA 92128/CALL (619) 451-0151

Circle 135 on Reader Service card.

create screen fonts for every PostScript font from just the manufacturers mentioned, we would be hopelessly swamped. It is our hope to soon make available a utility that will allow Amiga users to create surrogate screen fonts so that they can access the many fonts that are resident in typesetters and laser printers. This would give Amigan's even better access to the power and beauty of font libraries from companies such as ITC, URW and Agfa CompuGraphic. But until these corporations are made aware of the power and size of the Amiga marketplace, and release their libraries in Amiga-compatible format, no Amigan will truly have full access to such fonts.

Mark Wetzel,
Soft-Logik Publishing Corporation
St. Louis, Missouri

Dear AC:

Thanks for an exceptionally interesting and informative May 1989 issue! I am considering starting a business that will rely heavily on the Amiga's music and video capabilities, and this issue gave me a real boost in the motivation department!

I was distressed to read John Steiner's "Bug Bytes" report about the infected fake version of VirusX (known as Vx33.arc) that has been making its rounds on the BBS's. It's bad enough when some idiot creates a fake version of ANYTHING and tries to pass it off as an upgrade of a well respected program. But to intentionally create and distribute a program CONTAINING a virus that users are expecting to REMOVE viruses—that's the lowest form of low I can think of.

Steve Tibbett's VirusX is a godsend. I have kept up with the upgrades and have installed it in the startup-sequence of nearly every program I own. VirusX has saved me a lot of problems and Tibbetts is to be commended for a fine programming effort.

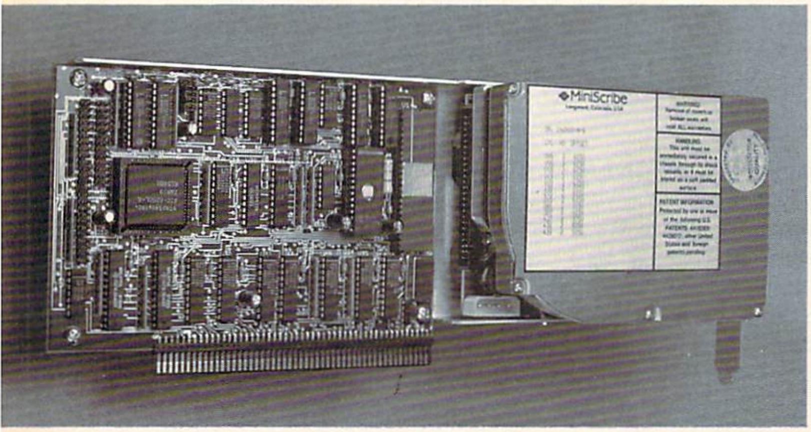
I hope that people will ignore the fake version and not shy away from the real VirusX. Don't help to destroy the future of one of the most important utilities in the Amiga community. Let's stand behind Steve Tibbetts and show him that we appreciate and support his efforts. Instead of downloading the infected fake from BBS's, lay out the $6 for a Fred Fish disk containing VirusX. That way, you know you are receiving a genuine version. More importantly, you'll be showing Steve Tibbetts that his efforts have not been in vain.

Irene C. Kobelski
Colchester, CT

-The latest version of VirusX is V3.2 and is available on Fred Fish Disk #216. Steve Tibbetts is doing a great job at keeping up with viruses as they come out.—ED

HardFrame/2000

The *Super-Speed*, DMA, SCSI Hard Disk Interface for the Amiga® 2000



- **AutoBoots AmigaDOS 1.3 *(Price Includes HardFrame Eprom!)***
- ***Directly* Boots the New Fast-File System! *(Doesn't Need Old FS!)***
- **Auto-mounts All Hard Disk Partitions *(no Mount List Required!)***
- **Designed-in, Ultra Strong, Multitasking Performance**
- **High Quality Metal Frame for Stable, On-Card, Hard Disk Mounting**
- **Power Cabling Directly from Card to Disk**
- **50-pin Cable Included**
- **Supports up to *seven* SCSI hard disks of any size**

How fast is *fast*? **HardFrame/2000** transfers data at Amiga bus speeds! It's actually faster than the hard disk mechanism itself! And even more important in the Amiga's multitasking environment, **HardFrame/2000** has extremely efficient DMA circuitry to get on and off the bus in almost no time at all: 280ns to get on; 200ns to get off. And it's true, dedicated DMA, too! **HardFrame/2000** autoboots *and* automounts *directly* into the AmigaDOS™ 1.3 Fast File System (old file system partitions are *not* needed!). The core of any DMA SCSI interface is in its SCSI protocol chip and DMA chip. MicroBotics has chosen the new, high performance Adaptec AIC-6250 SCSI chip, capable of up to 5 megabytes per second raw transfer speed, and the Signetics 68430 DMA chip running at 12.5 megahertz. Then we added additional FIFO buffering and enabled 16-bit wide data transfers for maximum throughput. The sophisticated design of **HardFrame/2000** provides for automatic SCSI arbitration, selection and reselection. The hardware supports either synchronous or asynchronous data transfer. **HardFrame/2000** can function as either the SCSI bus initiator or the target and can reside in a multiple master environment. Physically, **HardFrame/2000** is optimally flexible: the compact, half-size card comes attached to a full length, plated aluminum frame. The frame has mounting holes positioned to accept standard, 3.5" SCSI hard disk units such as those manufactured by MiniScribe, Seagate, Rodime, and others (hard disk mechanisms must be supplied by the user or his dealer as a separate purchase item). Alternatively, you can cable-connect to a SCSI drive mounted in your Amiga's disk bay or in an external chassis. As many as seven hard disks may be connected to a single **HardFrame/2000**. There is no size limit on each disk. **HardFrame/2000** includes a 50-pin SCSI cable and header connectors for either 50-pin or 25-pin cable connection. Also included is a current tap to power frame-mounted drives directly from the slot itself. **HardFrame/2000** comes complete with driver, installation, and diagnostic software.

Available NOW! Suggested list price, $329 (hard disk not included)
Frameless version: $299.00. See your Amiga Dealer.

The HardFrame/2000 photo shows the product with a MiniScribe twenty megabyte hard disk installed. Hard disks are *not* included in the purchase price of HardFrame. Note that if placed in the first slot, HardFrame uses only one slot even with a disk attached.

New!

8-UP! (DIP) FastRAM

Another great memory board from MicroBotics, 8-UP! (DIP) is the "brother" of the original 8-UP! (which uses SIMMs and PopSIMMs to fill its memory space). 8-UP! (DIP) uses conventional 1 megabit RAM chips in standard sockets to provide your Amiga 2000 with 2, 4, 6, or 8 megabytes of autoconfiguring FastRAM! 8-UP! (DIP) is a super efficient CMOS design for lowpower consumption and high reliability. Suggested list price, $239 (0k installed)

Join MicroBotics
ONLINE TECHNICAL SUPPORT
CONFERENCE ON BIX
(The Byte Information Exchange)
–call 1-800-227-2983
for BIX membership information!

MicroBotics, Inc.
Great Products Since the Amiga Was Born!
811 Alpha Drive, Suite 335, Richardson, Texas 75081 (214) 437-5330

Tell your dealer he can quick-order from MicroBotics directly – no minimum quantity –show him this ad!

"Amiga" is a registered trademark of Commodore-Amiga. "HardFrame/2000", "8-UP!", "PopSimm", are trademarks of MicroBotics, Inc.

Circle 109 on Reader Service card.

Amazing Users Unite

CAUSe
Canberra Amiga Users Society

John Buttle, Public Relations Officer
P.O. Box 596
Canberra City A.C.T. 2601
Australia
Phone: (062) 514141 (home)

Description: Our group has been operating since October 1986 and has approximately 200 members.

Meeting: Second Thursday of every month at 8:00 p.m.

Location: Seventh Day Adventist Church Hall, 3 Macleay St, Turner.

BBS: Phone (062) 551469, 24 hrs., sysop: Peter McNeil: (062) 545545 (home).

Users Unite Users Unite

O.P.A.U.G.
Olympic Peninsula Amiga Users Group
Eldon DeHart
717 E. 9th Street
Port Angeles, WA 98362
Phone: (206) 452-6735

Meeting: Last Friday of every month at 7:00 p.m.-?

Location: 717 E. 9th Street Port Angeles, WA 98362.
BBS: The Red Dragon, 2400 Baud, 24 hrs., phone: (206) 457-3671, sysop: Eldon DeHart.

Users Unite Users Unite

Twin Ports Area Amiga Users Group

Kenji Ogura
30 W. Superior St.
Duluth, MN 55802
Phone: (218) 722-2641

Meeting: Second Tuesday of every month at 7:00 p.m.

Location: UMD, Room MWAH 176.

Club Amiga
Joyce Burek, President
10002 Kirkglen Drive
Houston, Texas 77089
Phone: (713) 481-4704

Club Amiga meets on the first Thursday of every month at 7:00 p.m. at MicroSearch Computers, 9896 S.W. Freeway, Houston, TX 77074.

The Clear Lake Chapter of Club Amiga meets on the third Thursday of every month at 7:00 p.m. at MicroSearch, 2402-F Bay Area Blvd., Houston, TX 77058.

The Willowbrook Chapter of Club Amiga meets on the third Thursday of every month at 7:00 p.m. at Software Etc., in the Willowbrook Mall, located at FM 149 & FM 1960, Texas.

BBS: 1200-2400 Baud, 24 hrs., phone: (713) 528-7511, sysop: Scott Denham.

Users Unite Users Unite

WAC
Washington Area Commodore User Group
John Krout, Amiga SIG Leader
P.O. Box 684
Springfield VA 22150-0684
Phone: (703) 273-6074

Description: Serving Amiga users as well as C64, C128 and MS-DOS users in the region around Washington, D.C. The public is cordially invited to attend all of our meetings. We offer a 24-page monthly newsletter, an Amiga PD/shareware Disk of the Month, and a video library. Our club is especially eager to receive demo disks and instructional videos from software publishers. These items are publicized prominently in our newsletter.

Meeting: Third Saturday of every month at 1:00 p.m.

PACE
Penobscot Amiga Computer Enthusiasts

Michael Weinberg, Secretary/Treasurer
27 Cedar Street
Belfast, Maine 04915

Description: PACE meets once a month, maintains a PD library of more than 200 disks, and publishes the newsletter PACER every two months.

Meeting: Contact Michael Weinberg for dates, times, and locations.

Users Unite Users Unite

A.C.E.
Amiga Computer Enthusiasts

John Zale, President
P.O. Box 591
Oak Forest, Illinois 60452
Phone: (312) 687-7788

Description: We currently have 75 active Amigaists. The group has a wide variety of interests and experience. We welcome all those who have an Amiga or are curious about this wonderful machine to visit our next meeting.

Meeting: First Thursday of every month at 7:30 p.m.

Location: Percy Hopkins Auditorium at Christ Hospital, 4400 W. 95th Street, Oaklawn.

Users Unite Users Unite

If you would like your users group info printed here please send us a letter.

Thanks •AC•

The Amazing Computing Freely Redistributable Software Library announces the addition of...

New Orleans Commodore Klub's

inNOCKulation Disk

Version 1.5

To help inform Amiga users of the newer Amiga viruses and provide them with the means to detect and eradicate those pesky little critters!

Files and directories on the inNOCKulation Disk include:

Virus_Texts (dir)
Various text files from various places (Amicus #24, PeopleLink, and elsewhere!) describing the Virus(es) and people's experiences and their recommendations; TVSB "The Virus Strikes Back": satirical text describing future efforts to rid the universe of the dreaded (silicon) viruses! Interview with the alleged SCA virus author!

WB_VirusCheckers (dir)

VirusX3.2
Runs in the background and checks disks for viruses or non-standard boot blocks whenever they are inserted. (Recognizes several viruses and non-standard boot blocks. Removes virus in memory. Has a built-in "view boot blocks" & other features.)

Sentry
Revision of VirusX1.01 in Lattice C.

ViewBoot
Highly active mouse-driven disk and memory virus-checker which allows you to look at the pertinent areas (useful in case you suspect a NEW virus!)

VRTest3.2
Watches memory for viruses; will alert the user and allow their removal if found. Can check & INSTALL disks, etc.

CLI_VirusCheckers (dir)

AVirusII
From The Software Brewery (W. German). Disables a virus in memory.

Clk_Doctor3
Corrects problems with the clock (caused by malignant programs, perhaps not really a "virus") (A500 & A2000)

Guardian1.1
Checks for attempts at viral infection at boot! Allows you to continue with a normal boot (if desired). Includes a small utility program to permanently place the program on a copy of your Kickstart disk.

KillVirus
Removes (any?) virus from memory.

VirusKiller
A graphically appealing and user friendly program by TRISTAR.

Boot-Block_Stuff

SafeBoot2.2
SafeBoot will allow the user to save custom boot sectors of all your commercial disks and save them for such an emergency. If a virus somehow manages to trash the boot sectors of a commercial disk, just run SafeBoot and it will restore the boot sectors, therefore saving your disk!!

Virus_Alert V2.0.1
Yet another anti-virus program with a twist. Once installed on your boot disk a message is displayed just after a warm or cold boot notifying the the user that the disk and memory are virus-free, and forcing a mouse-button press before continuing.

BootBack1
Saves and restores boot-blocks. Runs from CLI only.

Antivirus *aka AVBB*
Includes SEKA assembler source.

XBoot
Converts a boot-block into an executable file, so you may use your favorite debugger (Wack, Dis, ...) to study it.

The inNOCKulation disk also includes icons and arc files.

To order the inNOCKulation disk, send:

$6.00 *includes postage & handling*

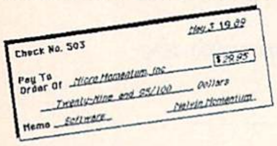
($7.00 for non-subscribers)

Amazing Computing
inNOCKulation disk orders
P.O. Box 869
Fall River, MA 02722

micro (mī'krō) enlarging or amplifying.
momentum (mō men'tem) strength or force that keeps growing.

- Webster's

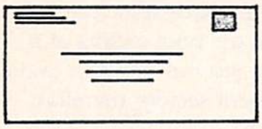
Momentum Check


Check No. 503
May 3 19 89
$29.95
Pay To Order Of Micro Momentum Inc
Twenty-Nine and 95/100 Dollars
Memo Software
Melvin Momentum

$29.95

A full-featured checkbook management package that makes checkbook management easy. Class codes are user-defineable and allow you to track any expense you wish. Use the standard reports provided or create your own custom reports. Reconciliation was never so easy. *Momentum Check* prompts you for all outstanding items! You can use the mouse, keyboard, arrow keys, or a combination to move around in *Momentum Check*.

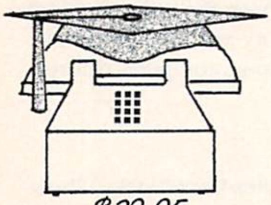
Momentum Mail



$29.95

An easy-to-use mailing list management program. Why fiddle with 300-page manuals and spend several hundred dollars when it can be as easy and affordable as this? *Momentum Mail* comes ready to use, waiting for your mailing list! Keep mailing lists for Birthdays, Christmas, Church, etc. You can search by any field and use pattern matching. User-adjustable label size.

TeleTutor



$29.95

A complete interactive telecommunication tutorial. Everything about telecommunications in one place! Features a simulated BBS to get the feel of using a BBS before going on-line. Practice uploading, downloading, using file compression utilities. Comes with a terminal program to get you on-line. You can save each lesson and start where you left off later. Complete listing of BBS numbers for U.S. and Canada. Start telecommunicating today!

Uzzi Interface

$34.95

A joystick/mouse interface with an auto-fire rate adjustable up to 30 rounds per second!! Switch between automatic and transparent mode. A 4 ft. extension cable allow easy connection for A500 owners. Adjustable fire rate allows you to fine tune the *Uzzi Interface* for each of your favorite games. Blow your old game scores away! Designed on and for the Amiga.

All our products are designed, produced, and manufactured proudly in the U.S.A.

Available at fine dealers, or order direct. Make check or money order payable to:

Micro Momentum, Inc.
100 Brown Avenue
Johnston, RI 02919
(401) 949 5310

Dealer Inquiries Invited

Please add $1.50 for S&H. C.O.D.s add additional $2.50. All products 90-day warranty.

If you've got a product, we're interested. Momentum Check, Momentum Mail, TeleTutor, and Uzzi Interface are trademarks of Micro Momentum, Inc. Amiga is a registered trademark of CBM.

Circle 125 on Reader Service card.

New Products and Other Neat Stuff

by Elizabeth G. Fedorzyn

Tune in

MicroIllusions has introduced **Music-X**, a music processor for use with the Amiga and MIDI instruments. Music-X works to create and manipulate musical scores, as well as communicate with and control MIDI devices.

Music-X supports real-time recording of systems exclusive data. Through the menu entries and gadgets on the Sequencer Page, you can record instrument segments one by one, edit them, then play them back together on MIDI instruments.

The Bar Editor displays events graphically as icons positioned in a two-dimensional time/pitch graph. Events can be added to, removed from, and repositioned in the graph. The Event Editor lists all events in a sequence, where they can be manipulated with functions like Cut, Copy, Paste, Recall, etc. The program even lets you record while in the edit mode, allowing you to watch notes appear on your edit display as you play them.

Music-X features keyboard mapping, allowing you to control almost any function of the sequencer—including starting/stopping the sequencer, initiating sequences, or changing the keymap itself—from a MIDI keyboard, footpedal, or other MIDI device.

Other features include the Librarian Page, which is used to store programs for synthesizers, and Patch Editor Pages, an expandable feature that allows you to work with different MIDI devices. Music-X comes equipped with patch editors for the Roland D-50, Yamaha TX-81Z, and Yamaha DX 1000.

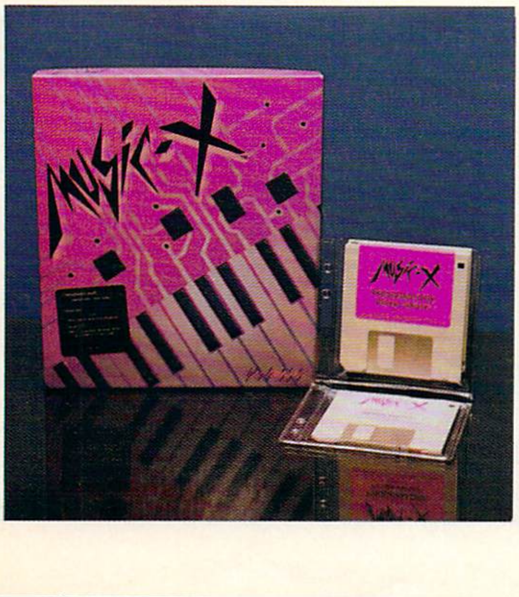
The package includes 1 program disk, 1 Music-X examples disk, 1 Music-X utilities disk, and a manual. Music-X is compatible with MicroIllusions' MicroSMPTE and MIDI-X. Music-X works with the Amiga 500, 1000, and 2000 and requires at least 1 meg RAM.

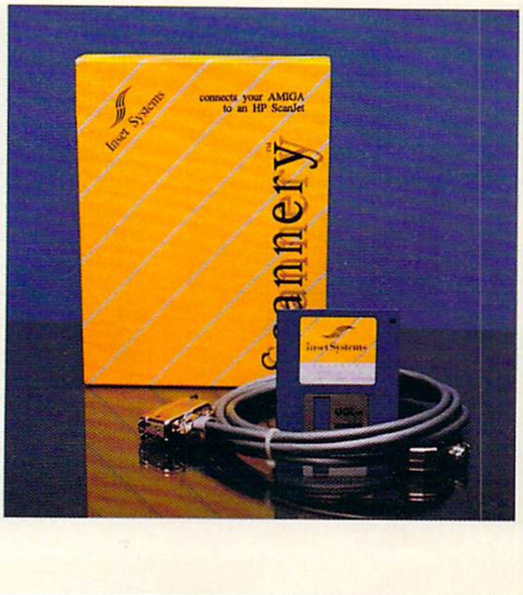
MicroIllusions
17408 Chatsworth St.
Granada Hills, CA 91344
(818) 360-3715; in CA (800) 522-2041
Music-X, $299.95
Inquiry #269

The missing link

New from Inset Systems, **The Scannery** links your Amiga to an HP ScanJet to give you high-quality scanned images for your documents.

Simply place your graphic image—document, chart, line drawing, photograph, etc.— face-down on the scanner bed. Select Preview from the options list,





Left:
MicroIllusions' Music-X

Right:
The Scannery from Inset Systems

and you can scan and view at a low resolution an entire 8 1/2 x 11 image on the left side of the screen.

With the Clip option, you can isolate any area within the image for the final high resolution scan. Just place the cursor in any corner of the area to be defined, and click and hold the select button while you drag the cursor to form a box of any size. The clipped area can now be fine-tuned by specifying the dimensions in inches or pixels.

The Scannery lets you customize your images, too. Once the dimensions of the outline are defined, move the box around the image field to precisely capture the elements as you want them to appear in the final scanned image. You can crop a picture to fit precisely into the space allotments of your document. Scale, Resolution, and Brightness can also be manipulated for greatest flexibility in image output.

The program lets you save in up to 16 shades of gray, or select one of four dither patterns for your output. You can also invert the image in both the Preview and Final Scan functions. Images can be saved to disk for further revision in a paint program, or for inclusion in a desktop publishing application.

The Scannery features ARexx support. The package includes the program disk, custom cable, and documentation. It supports IFF, EPS, PCL, and FAX file formats. The Scannery works with the Amiga 500 or 2000, and is compatible with the HP ScanJet or ScanJet Plus.

Inset Systems, Inc.
71 Commerce Drive
Brookfield CT 06804
(203) 775-5866
The Scannery, $250.00
Inquiry #271

Qui!, Quoi!, Quand!, Où!

Blue Ribbon Bakery's quest for organization has taken a international twist. Their organizational program, **Who!, What!, When!, Where!**, will soon be available in German and French language versions.

Beginning this month, the German version of WWWW will be available in Germany, Switzerland, and Austria. In December of this year, the program's French language edition will be released. (Never again will the French have to worry about missing a one-night-only showing of the *Nutty Professor.*) The French version will be made available in France as well as the Canadian province of Quebec.

WWWW acts as an electronic secretary, its many features including appointment scheduler, calendar, alarm clock, and Rolodex.

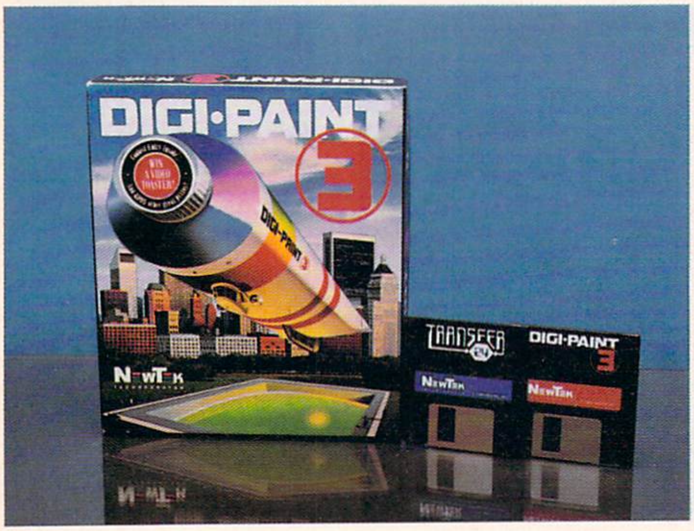
Blue Ribbon Bakery
1248 Clairmont Road, Suite 3D
Atlanta, Georgia 30030
(404) 377-1514
Inquiry#272

What happened to 2?

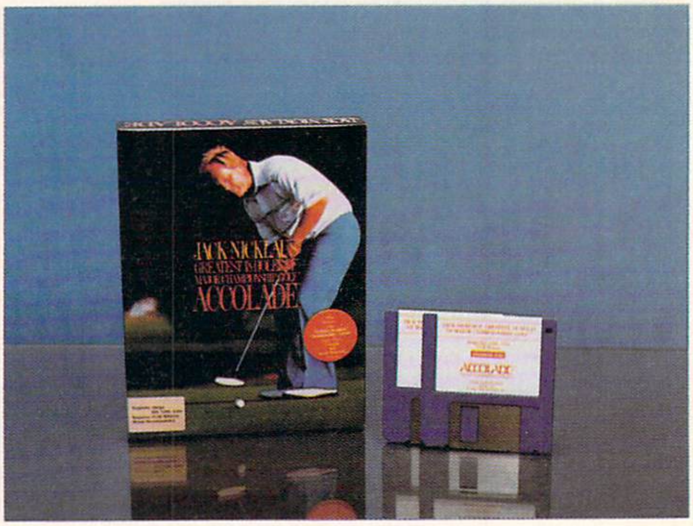
NewTek didn't waste any time with Digi-Paint 2; they've hopped right in with **Digi-Paint 3**, the "Ultimate Paint Program" for the Amiga that lets you create original artwork using all the machine's 4096 colors simultaneously.

The new program features, among other things, Texture Mapping with anti-aliasing for smooth 3D effects, Transparency Control; Text Rendering for rainbow fonts, anti-aliased fonts, etc.; Colorized mode that lets you add color to any black-and-white image; cut and paste with any drawing tool; and real-

Digi-Paint 3
from NewTek



Jack Nicklaus' Greatest 18 Holes of Championship Golf
from Accolade



time auto-scrolling on super bitmaps.

Digi-Paint 3's interface was implemented by renowned Amiga artist Jim Sachs. It features complete keyboard-control options and expandable palette selection tools.

Digi-Paint is compatible with ARexx and X-Specs 3D, and requires 512K memory. The package includes 1 program disk, 1 Transfer 24 image manipulation disk, and manual.

NewTek, Incorporated
115 W. Crane St.
Topeka, KS 66603
Digi-Paint 3
Inquiry #276

The Golden Bear stalks the Amiga

The latest golf simulation by Accolade finds golf's Golden Bear on an unfamiliar course—the computer screen. But he appears to have adapted to the electronic green rather well.

Jack Nicklaus' Greatest 18 Holes Of Major Championship Golf was jointly produced by Accolade and Jack Nicklaus Productions, Inc. The game, which was nominated for Best Sports Program of the year by the Software Publisher's Association, features graphically detailed representations of 18 of the world's most challenging holes in professional golf—all chosen by Nicklaus himself.

Some of the more prominent courses include Pebble Beach, Augusta, and Royal Lytham. Also included are Castle Pines in Colorado and the Cochise Course at Desert Mountain, Arizona—two of the toughest and most scenic courses Nicklaus ever designed.

Greatest 18 Holes incorporates a wide variety of challenging features including wind direction and intensity, course hazards, rolling hills, random pin placement, and skins scoring or stroke play.

Options allow you to play with one to four people or against a variety of computerized men and women golfers. You can play at either the beginner or expert level, on the pro, mens' or ladies' tees. Each computerized golfer has their own unique playing style and appearance (different plaid patterns?). If you're feeling particlularly handy with the irons, you can play against against a computerized Jack Nicklaus who possesses skills patterned after the golfer's game.

According to Accolade vice president Ralph Giuffre, the pro golfer's expertise was incorporated not only into the game's design, but is also extended to the user through both on-screen and documentation tips from Nicklaus.

This is the first computer software product release to be undertaken by Jack Nicklaus Productions. Commenting on the integrity of the release, Nicklaus said, "I'm told The Greatest 18 Holes of Major Championship Golf is the most realistic golf simulation ever made for the computer, and I believe it. I think you'll find it visually attractive and a delight to play. You definitely will find that, like the real game, it requires strategic thinking, concentration, and plenty of practice to master."

Jack Nicklaus' Greatest 18 Holes works with the Amiga 500, 1000, and 2000 and requires 512K memory.

Accolade
550 S. Winchester Blvd., Suite 200
San Jose, CA 95128
(408) 985-1700
Jack Nicklaus' Golf, $49.95
Inquiry #270

Get under your skin

Optimum Comparitive Technologies has introduced the **Opticomp** system, the first image-analysis system based on the Inferred-Structure Technology (IST).

The Opticomp system actually allows one to see under the skin of a human being, into the outer surface of the structural component of a person, or the myofascial body, without radiation or invasive techniques. Such a process allows structural physicians to make observations concerning ailments that lie beneath the skin.

Unlike X-rays, CAT-scans, NMR's, and other radiation-based treatments, that see all the way to the bone, Opticomp and IST lets physicians view the soft tissue and fascia, thus allowing them to discover the *cause* of such structural problems.

The easy-to-use Opticomp software handles all the views neccesary, files the views as sessions, and processes the images for a variety of analyses. The program can be operated via the keyboard or mouse.

The Opticomp system is sold as a complete turn-key package, including an Amiga 2500, a high-resolution color monitor, high-resolution color printer, custom digital signal processor, camera, custom lights, booth, and installation and operation manuals.

Optimum Comparitive Technologies
5306 Cromwell Court
Roanoke, VA 24018
Opticomp, $9995.00
Inquiry #275

•AC•

Which
hard disks
for
AMIGA?
Curious?
Any
(IBM compatible)
with our **A.L.F.**!
(**A**miga **L**oads **F**aster)

Safer with CHECKDRIVE.
Faster with FASTFILE-SYSTEM.
50% more MB with RLL-CONTROLLER.
More economic – even defective hard disks can be used.
SCSI-Bus, ST412/ST506-Bus
Automount Ver. 1.6, Autoboot Ver. 2.0

Prespect Technics Inc.
P.O. Box 670, Station H
Montreal, Quebec H3G 2M6

Fax: (514) 876-2869

BSC Büroautomation GmbH
Postfach 400368
8000 München 40 W-Germany
Phone: (89) 308-4152
Fax: (89) 307-1714

Circle 133 on Reader Service card.

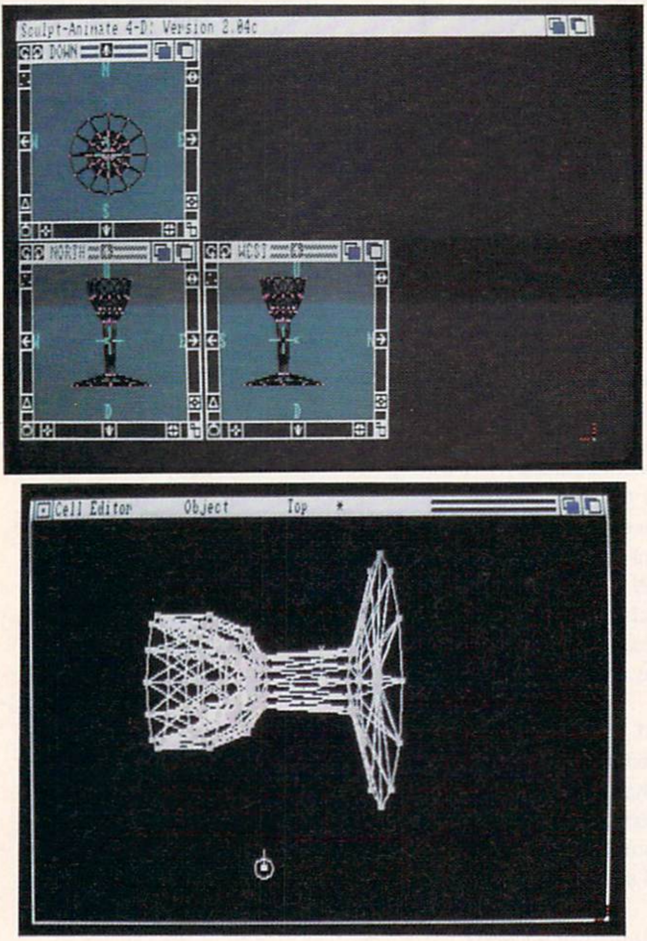
Secrets of Visual Synthesis

—Ray Tracing with Sculpt-Animate 4D and Turbo Silver 3.0

by David Duberman

We hold this truth to be self-evident: The Amiga is the most exciting graphics-oriented computer available to the average consumer. There are literally hundreds of ways to create images on the Amiga, and new ones are becoming available almost daily. However, many will agree that the most fascinating pictures come from ray-tracing programs.

Images produced with Sculpt-Animate 4D (top) and Turbo Silver 3.0 (bottom).



I call ray-tracing Visual Synthesis because of the analogy between this process and the creation of new sounds using electronic musical instruments. Just as the skillful programming of synthesizers can produce remarkable new tonalities never before heard, and perhaps better suited than traditional sounds for creating music in a rapidly changing world, ray-tracing—in experienced and insightful hands—can produce seamless photograph-quality images that can constitute a new art form.

Defined precisely, ray-tracing is a process whereby three-dimensional images and sequences of images or animations are generated by calculating the effects of rays of light upon a set of objects represented in the computer by vertices, faces and surface properties. By realistically depicting shadow and reflective effects, ray-tracing programs can generate startlingly lifelike images, with imagination being the only limitation on content and presentation.

Currently there are two major ray-tracing programs for the Amiga—Turbo Silver 3.0, from Impulse Inc., and Sculpt-Animate 4D, from Byte by Byte. Having gone through several major revisions since their original releases, both can be regarded as reasonably mature products. Each is superb in its own way and, if there is any way your budget can justify it, I recommend you purchase both.

But I realize that many with an interest in ray tracing derive little or no income from it, and can afford to purchase only one package. I will therefore present a list of similarities and differences between the two in an attempt to clarify the situation for the potential user of either or both.

The program environment

First we'll look at the physical program environment, starting with program size. While the Turbo Silver 3.0 program occupies approximately 250K of disk space, Sculpt-Animate weighs in at a hefty 460K or so—an important factor for those with the minimum requirement (for ray-tracing) of 1 megabyte of RAM. However, Sculpt-Animate uses an overlay system, with sections of the program being unloaded and loaded as needed so as to reserve the most possible RAM for object storage and image rendering.

One of the innovations in the 4D version of Sculpt-Animate gives you menu commands to load and unload program code (i.e., overlays). If the code is fully loaded, less access to the program disk is required, which is good for users of floppy-disk-based systems. But if the code is unloaded, an additional 270K of memory is available for other purposes.

Sculpt also lets you unload the Workbench to recover a valuable 30K of graphics memory, an option unavailable with Silver. Since Turbo Silver loads fully when initialized and doesn't use overlays, floppy-based users can remove the program disk after loading the program and use the disk drive for data.

Both programs come with free versions that support the combination of a 68020 microprocessor with a 68881 floating-point mathematics coprocessor—an expensive but necessary add-on for the professional ray tracer. While Turbo Silver is not copy-protected in any way, Sculpt-Animate has a form of copy protection: it often asks you for a word from the manual upon starting, which is rather annoying. Of course, price is also an important basis for comparison. While Turbo Silver's suggested list is $200, Sculpt-Animate's is $500.

Both programs use standard Intuition menus and include keyboard equivalents to many menu commands, as well as the ability to set up your own keyboard equivalents to any menu command. Silver lets you program the function keys alone and in combination with the Esc key for a total of 20 user-defined keyboard alternatives, while Sculpt lets you assign commands to the 26 letter keys.

> *"Ray-tracing—in experienced and insightful hands—can produce seamless photograph-quality images that can constitute a new art form."*

Sculpt also lets you assign menu commands to letter keys on-the-fly, but there is no way to save these definitions. Both programs also use requesters, but Silver uses convenient "Super requesters" that let you make a wide range of settings simultaneously. Overall, however, Sculpt's interface has a smoother, more professional feel and interferes less with the creative process.

Object creation

Next we'll compare each program's object-creation environment. The initial contrast is obvious. While Sculpt-Animate gives you the Tri-View (three movable and resizable windows showing views simultaneously from the top or bottom, front or back, and right or left side), Turbo Silver gives you a non-manipulable full-screen single view of the front, top, and right sides only. Also, the edges around the Tri-View's window contain a variety of useful gadgets, which, in many cases, obviates the need to resort to menu commands.

A major problem with using ray-tracing programs on the Amiga is that everything takes too long. It is well known that rendering images can consume exorbitant amounts of time, but image creation and manipulation can also involve lengthy waiting periods because of the time required to redraw a complex wireframe view whenever a physical change is made to an object.

In Sculpt-Animate's case, this problem is potentially tripled in magnitude because three windows are always redrawn. But author Eric Graham has provided an ingenious workaround. You can simply interrupt the redraw process to issue any other command—whether from menus or the keyboard—and it will immediately take effect.

Turbo Silver does not let you issue new commands while the screen is being redrawn after a transformation. On the other hand, if you abort a Sculpt-Animate rendering, there is no way to save the partial image. Turbo Silver lets you save any image interrupted during rendering.

The two programs let you create objects interactively point-by-point and face-by-face, but Sculpt also lets you define objects precisely with a list of vertices and faces entered into a text file called a script file.

Both programs also provide a means of creating spun (lathed) or extruded objects. While spinning is quite similar with both programs, Silver's extrusion is more advanced, as you can predefine a precise path and optionally force the extruded outline to "bend" with the path's direction. Sculpt's extrude makes a connected copy of selected faces and puts you in Grab mode, forcing you to manually define the size and direction of each section of the extrusion. On the other hand, Sculpt-Animate 4D offers a sophisticated helix command that lets you create complex twisted objects easily.

Also, both programs let you build intricate objects by linking a series of (user-defined) cross-sections. In this case, Sculpt's implementation (Unslice) is more sophisticated because the cross-sections need not contain the same number of points, as in Silver's Skin command.

Often, in creating three-dimensional scenes, you must rotate objects to reorient them with respect to one another. One of Sculpt-Animate's big advantages is that you can rotate objects interactively. In Silver, you must set the new angle on a separate requester while the object remains out of sight.

Fortunately, Silver's Undo command lets you instantly cancel changes that do not work out. Sculpt's lack of an Undo command is a serious deficiency. While both programs' Resize commands use a requester, Sculpt is more interactive, as you can see the object while resizing it.

Both programs let you move objects interactively, while offering different methods of selecting parts of objects for physical and/or surface manipulation. While Sculpt gives you a box-shaped Selector tool, Silver lets you use the keyboard to select contiguous sections of objects for transformation.

To a large degree, objects' surface properties define the nature of the ray-traced image, and there are significant differences between the two programs in this respect. Sculpt gives you a relatively limited selection of surface appearances—dull, shiny, luminous, mirror, metal and glass. Silver, on the other hand, gives you a sometimes dizzying array of choices. You can set an object's reflectivity and transparency independently to any of the Amiga's 4096 colors.

Other settings for the programs include specularity and hardness (both related to shininess), as well as roughness. You can apply IFF images to objects, optionally wrapping them around. Sculpt also lets you use IFF images, but only as backgrounds or foregrounds that are not really part of the scene. Another surface attribute unique to Silver is texture, a user-defined pattern (e.g., checker or brick) applied uniformly to an object. A special type of object in Silver is the stencil, a two-dimensional object defined externally in a one-bitplane (black and white) IFF file—the image area becomes the object.

As mentioned, Sculpt allows only one type of transparent object—glass. While this simulates the surface appearance of glass very well, it does not bend light the way real glass does. Silver lets you set an index of refraction for a transparent object, which causes the light to bend exactly as in reality. But even with refraction, Silver does not give you the highly glossy appearance of Sculpt's glass.

Both programs permit a practically unlimited number of light sources. While Sculpt's light is a special type of object, in Silver, any object can be a light source. Such objects, however, remain point sources of light—you cannot make a fluorescent tube by setting a cylinder to be a light source. Sculpt's lights always diminish in intensity with distance, but Silver gives you an optional sun-type light that lights the same regardless of distance, which is ideal for objects moving toward or away from the viewer. Both programs only depict shadows—a time-consuming activity in full ray-trace mode. But all of Sculpt's light sources cast shadows, while Silver makes each light's shadow-casting ability optional. As for "global" settings regarding ground and sky, an infinite plane "ground" is available in both cases.

Sculpt gives you a single optional ground pattern, a checkerboard. Silver's ground is blank by default, but you can apply textures or IFF images to it. The latter repeat automatically in a checkerboard pattern. With the sky, both programs permit a solid sky, graduated between two colors, or none. However, Silver lets you set the degree of dithering in graduated sky and offers an optional "fog" mode that obscures the distracting horizon line.

Finally, I cannot leave out Sculpt-Animate 4D's new font capability, a must for anyone creating animated three-dimensional titles. While libraries of 3D fonts are available for both programs, creating good-looking words in Silver demands a certain amount of experimentation with repositioning letters. Using fonts set up in the proper format for Sculpt requires you to enter words from the keyboard that are loaded from disk and lined up automatically as you type. Another of 4D's innovations useful for precision work is a variable 3D grid with snap capabilities.

Rendering

After creating an object, you must render it to view it as a three-dimensional solid. Both programs offer a variety of rendering modes, but Sculpt-Animate makes it easier to switch modes—it's a simple menu choice that can be made at any time.

Turbo Silver, on the other hand, forces you to close the current file before setting a new resolution. Thus, to render a scene in a different graphics mode, you must:

1. Write a cell's contents to a disk file.

2. Close the file.

3. Set the new graphics mode. Available choices are HAM/Lo-res/Hi-res, Interlace/Non-interlace, Normal/Overscan, and 12-bit/24-bit RGB, the latter being useful only with special display hardware.

4. Open a new file.

5. Load the cell and render it.

While Sculpt-Animate lets you make all these choices without unloading data, both programs let you choose full ray-trace or solid model/scanline mode internally. Incidentally, this latter compromise mode, which offers some of the benefits of ray tracing, including smooth surfaces with greatly improved rendering times, seems to be faster in Silver. Sculpt-Animate also offers paint and sketch modes that render objects as solids very quickly in lo-res or hi-res with no smoothing or shading.

Sculpt-Animate generates images in standard IFF format, while Silver uses a custom 12-bit format called RGBN in all modes. You can, however, save a Silver image in IFF format by pressing the S key while the image is displayed with the Show command. Consequently, converting a long sequence of Silver images to IFF format can be a lengthy and tedious process.

Of greatest importance to floppy-disk users might be the fact that Sculpt-Animate renders in memory, while Turbo Silver renders to disk. The process used by Silver is this:

1. The image is rendered until a small buffer is filled. Depending on image complexity, this can be anywhere from one to several scan lines.

2. The image file is opened.

3. The buffer data is written to the image file.

4. Return to Step 1 until the image is completed.

As you can see, there is apt to be quite a lot of disk activity involved in rendering an image in Silver. With complex images rendered in Solid Model mode, the process may be repeated several times per minute. When rendering to a hard disk or RAM: disk, this is not a problem. But when rendering to a floppy disk, due to current limitations of AmigaDOS and the inherent slowness of the medium, the process can be prolonged significantly.

On the other hand, if you are working in a limited memory situation (i.e., 1 to 1 1/2 megabytes of RAM), Silver's method lets you squeeze out every last available byte for object data. Since Sculpt-Animate renders in memory (and requires more memory just to run), you must restrict object complexity more severely. Also, to retain a rendered image with Sculpt, you must be sure to save it to disk.

> *"The two programs let you create objects interactively point-by-point and face-by-face...If your budget can justify it, I recommend you purchase both."*

If you are in a big hurry to see your picture, the programs offer markedly different compromise solutions. With Silver, you are always rendering at full size, but you can use the Set Zone command to define a rectangular area of any size for rendering—the rest of the screen remains blank.

Sculpt, on the other hand, lets you render the full image at a range of sizes from Overscan to postage stamp. Sculpt displays the image as it is rendered, while Silver makes you wait until rendering is finished before you can see it (actually, you must load the image file first).

A new feature with Sculpt-Animate 4D, now the default condition, is that you can delay displaying the picture until it is ready to be drawn completely, which decreases total rendering time.

AmiEXPO

THE

AMIGA

PERSONAL COMPUTER SHOW

Come See The California Goldrush!

October 20-22, 1989

Santa Clara Convention Center

Santa Clara, CA

Over 10,000 Attendees and 120 Amiga Companies Will Be There.
STRIKE IT RICH AT AmiEXPO-CALIFORNIA!

Admission includes the Exhibition, Seminars, Keynotes & Amiga Artists Theatre!
120 Amiga Exhibitors Featuring State of the Art Software and Hardware, at the lowest prices!

Master Classes Available in Amiga Graphics, Video, Programming, Animation, Music and Publishing!
Seating for Master Classes is limited; call for schedule and availability before registering.
PRE-REGISTRATION DEADLINE IS OCTOBER 6, 1989

For Hotel Reservations Call the DoubleTree Hotel at (408) 986-0700.
Hotel reservations deadline: September 27th. 1989.
For discounted airfares, call American Airlines at (800) 433-1790 and give them this ID: S-83536.

REGISTER TODAY!

Register by Mail, or Call 800-32-AMIGA Nationwide (or 212-867-4663)
For Your Ticket to The Amiga Event!

Circle 159 on Reader Service card.

Yes, I want to come to AmiEXPO - California

___ **Friday** ___ **Saturday** ___ **Sunday**

Registration is $5 Additional At The Door

One day - $15 _____
Two days - $20 _____
Three days - $25 _____

Master Class(es) - List Class and Time - $60 Each

________________________ _____
________________________ _____
________________________ _____
________________________ _____

Total Amount Enclosed _____

NAME ________________________
COMPANY ________________________
ADDRESS ________________________
____
CITY ____________ STATE ______ ZIP ______
For ___ MasterCard or ___ VISA Payment
Expiration Date ________________________
Account Number ________________________
Name as it appears on card: ________________
Signature ________________________

Make Check or Money Order Payable to:
AmiEXPO 211 E. 43rd St., Suite 301
New York, NY 10017

And as mentioned, Silver lets you save a partially rendered image, while aborting a rendering in Sculpt automatically discards the picture.

One of the most crucial criteria for comparison is that of image quality, and there are significant differences between the two in this area. The single word that best sums up Sculpt's most attractive images is "glossy". Sculpt's glass and metal surfaces have just the right degree of shininess and are truly convincing—qualities fairly difficult to obtain in Silver.

However, Silver is quite capable of producing stunning images and animations, and it is the only choice for those requiring a broad range of surface characteristics. Of course, you (or your clients) are the ultimate judge of image quality, so you should preview a wide range of images produced with both programs before making a decision.

Animation

While the two programs take a basically similar approach to animation generation, there are still marked differences between the two.

To create an animation, a sequence of still images is generated. The stills are then compressed and combined into a single animation which can be shown outside the program. Thus, if you are generating many frames, you may need a lot of disk space.

Both programs have the space-saving option of deleting frames as they go, but if the generation process is interrupted, you must restart from scratch. Each program uses its own proprietary format for animations. This format is not compatible with other graphics programs for the Amiga. Neither program provides internal sound capability. There are, however, a number of programs available which, given a sequence of IFF still images, provide sound-adding capability.

Both Silver and Sculpt allow global or path animation in which objects are assigned to follow linear paths during the course of the animation. Silver's paths are easier to use—the number of points used to define them bears no relation to the length of the animation, so you can set up a straight line animation of any length with a two-point path. Conversely, Sculpt's paths permit uneven motion, as each point on the path represents a subsequent animation frame and you can set points anywhere you like.

Both programs allow object rotation during path movement. Again, Silver's is easier to set up, but Sculpt's arrangement is more flexible. Still, Silver allows gradual resizing during path movement in any combination of the three dimensions. Silver also has a "Follow Me" command that lets you assign any number of objects to follow each other along a path with a single command.

If your Silver animations have many frames and complex objects, you had better have plenty of disk space available because the program uses a separate disk file to hold all animated objects in each frame. So, if you have 50 frames, each containing 60K of objects, you will need 3 megabytes of disk space just for the object data. You can set objects to be external, in which case they are saved only once, and can even be animated.

Sculpt's big animation extra is tweening, which is very important for animators trying to recreate natural motions, such as walking. With tweening, you set the beginning and ending frames of a motion, such as the stride of a walking person, and the program creates all the steps in-between. The term tweening comes from the early days of film animation, where animators would draw frames representing only the peaks of the action, and assistants would draw all the in-between steps, or "tweens". Sculpt lets you combine path and tweening animation in the same scene, but not with the same objects.

Silver offers relatively sophisticated animation script capabilities compared to Sculpt's limited ability to change only the timing of each frame, loop forward, or "Ping-Pong" mode. With Silver's movie script file capability (actually a primitive programming language), you can set up complex sequences from existing "exposed" frames, complete with "subroutines".

Other Movie commands let you pause the animation for a set number of seconds, change the frame rate, insert black frames for transitions, and wait for a keypress. Since one of Silver's great advantages is that it's much faster at compressing a sequence of exposed stills into an animation, it is easy to experiment with the Movie commands.

Summary

While I have gone to great lengths to point out the weaknesses of both Sculpt-Animate and Turbo Silver, I must again emphasize that I am an avid user of both programs and recommend them highly to all Amiga users interested in three-dimensional graphics. By all means, try to get a demonstration of either or both at your local Amiga retailer. If they prove cooperative, show your support by purchasing the program there, even if it does cost you a few more bucks than mail order.

•AC•

Product Information

Impulse Inc.
6870 Shingle Creek Parkway #112
Minneapolis, MN 55430
(612) 566-0221
Turbo Silver 3.0, $200.00
Inquiry #273

Byte by Byte
Arboretum Plaza II
9442 Capital of Texas Highway North
Suite 150
Austin, TX 78759
(512) 343-4357
Sculpt-Animate 4D, $500.00
Inquiry # 274

Tell Them You Saw Them.

When you contact a vendor, tell them you saw them in the magazine that delivers more.

Tell them you saw them in:

Amazing COMPUTING For The Commodore AMIGA
Your Original AMIGA Monthly Resource

PageRender 3D

review by Michael Fabrion

I am a professional graphic artist with twenty-five years experience. The Amiga was my first and only computer, and DeluxePaint my first software package. I chose the Amiga for its affordable graphics capabilities, and DeluxePaint because it was the only game in town. Three years and several paint programs later, I had still not gotten into 3D rendering. I had looked at other 3D programs, but their apparent complexity and lengthy rendering time scared me away.

Then Mindware International released PageRender 3D (Page3D), and it was a joy to use. It is quick and has a well thought out user interface that made it easy for me to comprehend. Page3D is an object oriented program which builds complex scenes or shapes from smaller objects, much like constructing a large building out of bricks and beams. Each object is built up of facets. The smallest objects, such as a square or triangle, contain only one facet. A cube is made up of six facets (each side is one square facet) and so on.

Page3D comes on two disks, a main disk and an extras disk. The main disk is self booting and contains the program and a large library of over eighty predefined objects. Page3D is not copy protected and can be easily installed on a hard drive. The extras disk holds a 68020/68881 version of the program for those of you lucky enough to be driving a turbo charged Amiga.

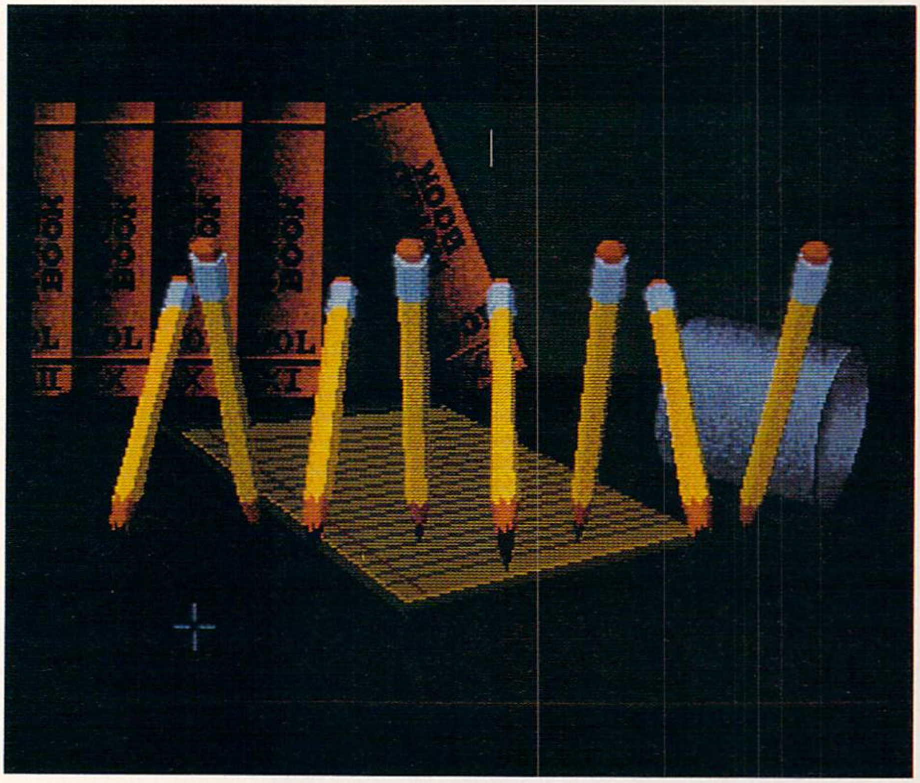
The rest of the space is stuffed with sample scripts that show off the numerous features of Page3D. These scripts are a great help in getting started with the program and much preferable to plodding through yet another tutorial in a manual. For those who want to get deeper into the third dimension, a pair of red/blue 3D glasses is included.

Yes, Page3D will work in a true 3D mode! And if the old red/blue method is not deep enough for you, Page3D supports the X-specs from Haitex Resources. The tools directory includes a module to trigger the X-specs when running a 3D Anim file with ShowAnim.

The 220-page spiral bound manual abounds with information and useful graphic examples. It could double as a guide to solid geometry. In the chapter Examples, many of the sample scripts contained on the extras disk are documented and explained. Included in the appendixes is a description of useful AmigaDOS commands, a comprehensive summary of ED and its commands for use as a script editor.

Getting Started

Clicking on the program's icon causes the Workbench screen to slowly scroll down off the monitor before Page3D appears. When the program loads, you are presented with two screens, the work area, where objects are displayed, and a small foreground screen that runs across the bottom. The bottom screen contains the tool gadgets and a small CLI input window. The default settings start you out looking down at the midpoint of an X, Y and Z axis, the center of Page3D's world. Clicking the gadgets enters commands for manipulat-



Below:
Writing utensils with rhythm—
"Dancing Pencils" produced with PageRender 3D

ing objects or your viewpoint within this 3D environment.

A slider to select the amount of movement extends under the gadgets. When you select a gadget, a visual marker on the work area shows you where and how far the object will move. The CLI window on the right echoes your mouse selections, both from the gadgets and the pull down menus, in Page3D's script language. I found this feature an immense help in learning how to use scripts.

Visualizing what a command does and seeing the script version simultaneously is an excellent teaching aide. The CLI window's curser is always active, so you can immediately enter commands from the keyboard. I soon found myself combining mouse and keyboard input, forgoing the slider for inputting the amount of movement. Page3D allows the user to define four macro keys for their own use.

As I learned my way around Page3D, sometimes I would get disoriented, but the different view commands (object, front, side or top) made it possible to find my way back. Page 3D can be mouse or keyboard driven; or you can use a combination of both. You can also write or generate script that the program will read and create while you watch.

Pulling up the foreground screen reveals a full window listing the commands you have been entering. You can select two other gadgets, SETUP and FACET. The SETUP gadget displays a detailed description of the current settings for screen resolution, page size, light source, observation point, type of movement, etc. You can configure Page3D to your liking and save the setup for future use. FACET displays a list of current objects in memory, the number of facets in each object and the object's position in relation to the X, Y and Z axis. The lower screen can be quickly moved up, down or flipped out of sight with a macro key.

Objects

Page3D comes with a library of over 80 objects, ranging from simple squares to many faceted spheres. It also includes the alphabet and a set of numbers. Most of these can be easily accessed from a pull down menu. You can load objects by selecting them from a requestor that scrolls up from the bottom, and you can import objects by entering "load filename" (with a full path to the file) from the keyboard.

For ease of movement, you can merge complex shapes created with smaller objects into one large object. Newly loaded objects from the Page3D library always appear with their center at the intersection of the three axes. When saved, new objects will retain their location in relation to the center of Page3D's world. They will reappear in that position when loaded. Objects have their own axis, so they can be rotated independently, as well as, around the program's X, Y and Z axes.

Adding to Page3D's versatility is the Create Object feature, which lets you create your own 3D objects in two formats, flat and cylindrical. Flat objects are rendered by drawing the outline as a series of connected, straight lines with the mouse and then selecting a depth for the third dimension.

Cylindrical objects are drawn as a cross section. You outline the shape on one side of a vertical center line while the program mirrors the opposite side. When the cross section is finished, entering the number of sides gives the program the data needed to render the new object. Objects can be saved for future use. Using the cylindrical method, my son quickly produced a flying saucer that he was able to send spinning and turning through space. The best method I found to create new objects was to design my object as a silhouette in a paint program, import it into Page3D, where the picture can be easily traced over for quick rendering.

Spaceship (top) and doll (bottom) produced with PageRender 3D

Colors and Resolutions

The default color of Page3D's objects is blue, but objects can be painted any color in the Amiga's spectrum. Not only can you have objects of different colors, but individual facets of an object may have distinct colors. It would have been helpful to match colors if you could "pick" the color of an existing object using the mouse. Currently, you must write down the color numbers used. If you don't keep track, you can end up with a lot of near-matching colors and a color spread that is too limited for effective shading.

While Page3D supports HAM, it does not support Extra Halfbrite. If you need to conserve memory or storage space, you can reduce the number of bitplanes. Although this decreases the amount of colors, the excellent dithering techniques provide the illusion that many more colors are being displayed than the palette can hold. Objects are scaled in centimeters and not by pixels, therefore changing resolutions does not effect the size of the objects on the screen.

Moving On

Objects can be moved in any direction (you are in 3D). They can be stretched, pulled and pushed into other shapes with the object editor. You can move freely around and through them. Movement of objects, view point and light source are determined in relation to the X, Y and Z axes. The commands View Front, View Side and View Top move you to a position looking straight down the X, Y or Z axes respectively.

The tool gadgets give you a choice of moving an object up, down, left, right, away from you, towards you or parallel to any of the three axes. Rotate lets you move the object around one of the three axes or spin on its own axis. An object's rotation is related to its center point and the axis it is spinning around.

The object's center point can be moved anywhere. By placing the center point outside an object, you can change its rotation into a swinging motion. Other tools let you move your observation point in any of the above ways or around the center point to the left, right, up or down. You can select Look to change your direction of observation. Additional object manipulation tools are clone, cyclic, size and mirror. Resizing can be done in any one direction,or overall.

Page3D can utilize any (or all) of four different systems of movement: Cartesian, Spherical, Cylindrical and Tetrahedral. Cartesian is the program's default and easiest to understand. It moves an object along or in relation to the three axes. Spherical moves objects around the surface of a sphere. Cylindrical utilizes a cylindrical shape that is straight up and down on the Z axis as a base. Tetrahedral follows the shape of a regular tetrahedron (a three-sided

Author, Author

A look at Michael Abrams, author of PageRender 3D

PageRender 3D Author: Multifaceted Individual

One might wonder what kind of person could construct a multifaceted program like PageRender3D. A multifaceted person, of course. That person is Michael Abrams. Michael's background finds him entrenched in both art and physics. In high school, he had an infinity for art—so much so that he was going to pursue it as a career. But then...

He changed his mind and earned his degrees in physics. He designed industrial machines for twelve years while dreaming of creating the ultimate 3D graphics program for the computer. Not having a computer, he began by figuring out his calculations on a hand-held calculator, then advanced to a programmable calculator. By the time Michael selected the Amiga as the machine to program on, he had good a portion of his work figured out. Of course, there was the hurdle of teaching himself C before he could create his program. The final version of PageRender 3D was the result of over 700 pages of programming.

The bulk of his program was done on his Amiga 1000 configured with only 512 on board and no hard drive. I first saw Michael's early version of PageRender 3D at one of our monthly Amiga Group meetings. For five dollars (the price of the disk and cost of the manual printout) he allowed members of the Cleveland Area Amiga Users Group to try out his program.

Being a pioneer at heart, I thought "Why not?", and plunged into the program. It wasn't very user friendly in those days, but luckily I am an artist who had a high school infinity for physics. Michael really listened to those of us hearty individuals who took learning the program on as a challenge. That was a year and a half ago, and it is amazing how many new features Michael has added since then.

Michael is one of those "what if" people who never stops thinking of the possibilities. Those of us hungry for good software that expands our machines' capabilities are happy about that.

pyramid). When you change to a different system of movement, the tool gadgets change correspondingly.

Animation and Scripts

Scripts offer the best method to create Anim files. All the movements of an animation sequence can be plotted and written into a script. "Save Script" will save to the destination file all the commands you enter with the mouse or from the keyboard. The commands can be edited later in any text editor. The "Save Anim" command can be included at the proper intervals. When your ready to create your animation, just select "Read Script" and go get a snack while the program renders the Anim file.

Backgrounds can be created in a paint program and loaded into Page3D. Setting the Autoclear to off causes the 3D object to be drawn over the imported background. But if you move an object with Autoclear off, the old image will remain. You must reload the background to clear the screen of the previous image.

This is no problem, as the Load command can be placed in the script. I copied my background picture to RAM and loaded it from there to speed things up and save on disk access. The palette used by PageRender 3D is set by the object files. You must first create your object(s) and save the screen as an IFF file, then load this image into the paint program before rendering the background to set the palette. Another approach is to import the Anim file created with Page3D into DPaint III. There you can create a background on the spare screen and merge it in back of the Anim scenes, and detailing can be added if you so desire.

Page3D can do simplified ray tracing. Surface textures cannot be defined nor are cast shadows plotted, but the increased delineating of light and surfaces can give spectacular results to the final picture. The effect is most noticeable in complex pictures with numerous small objects. The ray trace command can be embedded in a script file to produce amazing animations. Even this simplified ray tracing greatly increases the time it takes rendering a picture, especially in Hi-Res mode. When creating animation files, each frame will be ray traced before being added to the file, increasing the execution time significantly.

After several months of using PageRender 3D, in beta versions and the finished product, I've still not tried all the numerous features available. Some of them, such as the Tetrahedral movements, I have not yet begun to understand. There is much more then I've been able to cover in this review. It is much faster and simpler than most other 3D programs. However, you do sacrifice some of the detail that can be accomplished with other programs. For my purposes, I prefer Page3D's speed and ease of use. Page3D is a excellent program for those interested in getting into 3D rendering, either stills or animation.

•AC•

Mindware International
110 Dunlop Street, West
Box 22158
Barrie, Ontario, Canada L4M 5R3
1-705-737-5998
PageRender 3D, $159.00
Inquiry # 266

Amazing
Amazing
Amazing
AC #3
Amazing
AC #4

Amazing on Disk

Source Listings and Executables from the pages of Amazing Computing!

Only $6.00 per disk ($7.00 for Non-Subscribers)
Please use the order form on page CIII or the tear out form.

AC #1 V3.8 & V3.9
Gels In MultiForth Part I & II: Using Gels in MultiFourth.
FFP & IEEE: Math routines in Modula-2.
CAI: Computer Aided Instruction in AmigaBASIC.
Tumblin Tots: Save the falling babies a game. Written in assembler.
Extra Goodies: Three freely redistributable programs, VGad, MenuEd & Bspread.

AC #2 V4.4
Fractals Part I: An introduction to the basics of fractals with examples in AmigaBASIC, True BASIC, and C.
Shared Libraries: Using shared libraries in C.
MultiSort: Sorting and intertask communication in Modula-2.
Double Playfield: Using dual playfields in AmigaBASIC.
'881 Math Part I: Programming the 68881 math coprocessor chip.
Args: Passing arguments to AmigaBASIC.

AC #3 Vol.5 & Vol.6
Digitized Sound: Playing digitized sounds using Modula-2.
'881 Math Part II: Part II of programming the 68881 math coprocessor chip using a fractal sample.
At Your Request: Using the system supplied requestors from AmigaBASIC.
Insta Sound: Tapping the Amiga's sound from AmigaBASIC.
MIDI Out: A MIDI program that you can add to. Written in C.
Diskless Compiler: Setting up a compiler enviroment that doesn't need floppies.

AC #4 Vol.7 & Vol. 8
Fractals Part II: Part II on fractals and graphics on the Amiga in AmigaBASIC and True BASIC.
Analog Joysticks: Using analog joysticks on the Amiga in C.
C Notes: A small program to search a file for a specific string in C.
Better String Gadgets: How to tap the power of string gadgets in C.
On Your Alert: Using the system's alerts from AmigaBASIC.
Batch Files: Executing batch files from AmigaBASIC.
C Notes: The beginning of a utility program in C.

Inside the Making of

DeluxePaint III

An Interview with Dan Silva

by Ben and Jean Means

Every computer has one program that exemplifies its capabilities—Lotus 1-2-3 on the IBM, PageMaker on the Mac, and DeluxePaint on the Amiga.

What kind of people have helped create such an elegant user interface to delight the computer artist in us all? We wondered. So we put on our trench coats, strolled to the telephone and called our undercover informant at EA—Deep Ear.

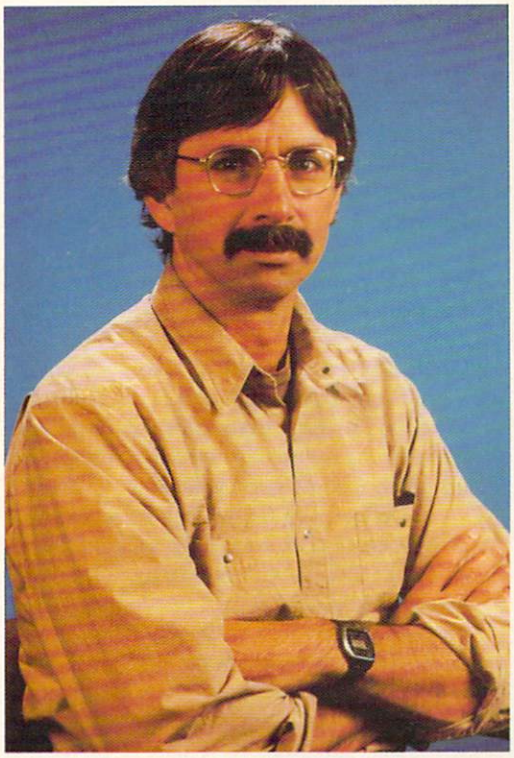
"Ya want Silva and the beta testers?" Ear yelped. "No way, Jose, yer askin' too darn much! Call me back when yer sober." I knew we had only moments before we'd hear his telephone handset crashing into its cradle. So I decided to call in a few old debts.

"Ear, hold on a second. Wasn't I the one who told you about the backdoor in Arkanoid?" "Yeah, but..." "Wasn't I the one who told you how to get the Babel fish in Hitchhikers Guide to the Galaxy?" He hesitated, "Sure, but..." I honed in for the kill. "Ear, you wouldn't want me to let it slip to the Bandito about your black market source for Obese Agnus chips, would you?"

"You wouldn't!" he snarled. "Try me," I chuckled. There was a long pause. "I'll call Silva and see if he can spare you some time," Ear snapped. We were in!

Getting our man

After graduating from Stanford with a masters in Mechanical Engineering, Silva worked for NASA Ames, Lucasfilms, and Xerox, where he entered the world of computer graphics with a black-and-white paint program called Doodle. When Silva and several others left Xerox in 1983 to form EA, Doodle ported its way onto the IBM PC, sprouted some new features and became Prism. When the Amiga arrived in 1985, Silva rewrote Prism to take advantage of the Amiga's chip set, and DeluxePaint was born—just one month after the Amiga's release.



A year later, after many nights of program-'til-you-drop intensity, DeluxePaint II arrived and created even more of a stir than the original. The "best of" awards poured in again, and Silva took a well deserved vacation. But the true glory of DP was yet to come and might not have come at all except for one fateful night of casual hacking. Here's how it happened.

AC: How did DeluxePaint III get its start?

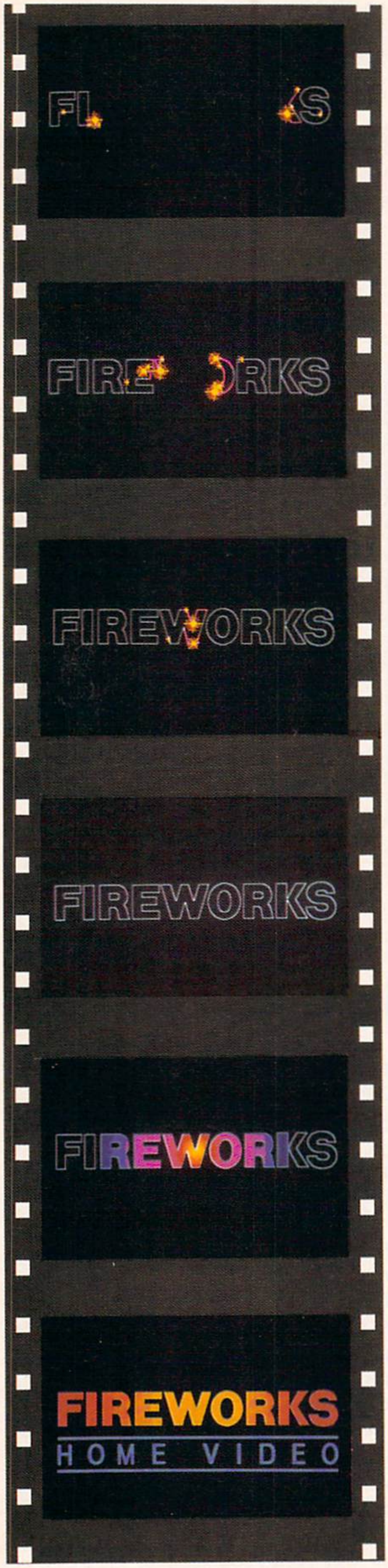
DAN: I'm naive, but I always end up getting myself into these things. Basically in one night of hacking, I created this page-flipping stuff, said, 'oh that was fun,' and ended up paying the price for over a year since then, trying to make it into an actual program.

It's the first step in putting together an idea I've had for a long time, which is simply that paint and animation can be viewed as two sides of the same thing. The same tools you use for paint can be used for animation, and vice versa. Paint is just animation where you don't erase, where you just paint on the same frame. I'd like to do more in that direction. This was just a first step in a quick experiment, which turned out to be a little more than I bargained for.

But I was feeling I had to champion the Amiga a little bit. We went through this incredible Amiga period at EA where it was just all we were doing, and then, frankly, we felt Commodore didn't carry the ball as well as we would have liked. The company shifted over toward the IBM and overreacted somewhat. I felt like, 'Hey! the Amiga is still a viable machine', and I wanted to keep supporting it. Since then we've shifted back towards the Amiga.

AC: Do you have a fine arts or graphics background?

DAN: No, but my Mechanical Engineering background from Stanford was definitely valuable training. The courses in M-E design gave me a way of thinking about programs as products that a lot of programmers don't have. When you think about a computer program as a complete product, it's different than just thinking of what features you can program. Every programmer should study product design if they plan to sell their programs. I also have a pretty strong background in math. Many



Left:
Fireworks anim title series created by DPaint III beta tester and graphics artist Kara Blohm

programmers don't understand the underlying mathematics, and that's too bad, especially in graphics programming where it's very important.

What got me started in computer graphics is that I really like painting pictures on the computer, and what I'm trying to do is make that process happen for myself. I was influenced by things I had seen on higher end graphic systems, but overall I just sat up nights to paint and implemented features so I could finish that painting. If something was too hard to do by hand, I would put in a feature to make it possible.

AC: Did you borrow any user interface tips from other programs?

DAN: Oh yeah, I'm always stealing any ideas I can. One of my rules is never be too proud to learn from someone else. Even a program that is basically crummy usually has some good ideas in it.

DeluxePaint III also had some real active beta testers, especially Jeff Bruette and Kara Blohm, whom gave me a lot of feedback about what they needed in order to do actual video production work. The beta testers were very important on this program because the animations take so much time to work with that I just didn't have time to be an artist testing it out and the programmer too.

AC: Did any features come from incessant whining from beta testers desperate to have a particular feature?

DAN: Yes, a lot of those came from Jeff Bruette (Laughs). He could probably tell you some of those. The font requester was from incessant whining, not just from Jeff, but from everyone. However, he was one of the main ones, because he has lots of fonts on his hard disk and wanted a better way of using them. And Kara made sure I got color fonts to work right, which I think is really good because it's a very powerful addition. People love to make titles to things, and Kara's color fonts just look great.

AC: How did the idea of the anim brushes come about?

DAN: That was motivated by the feeling that animation is hard and the average person isn't an animator, so it would be nice to provide an animation library of clip animation art for people to use. So the next question was okay, how do you create static clip art? With brushes, so the next step is to have an animated brush. Hopefully, that's going to enable a lot more people to create animations who don't feel like they can hand draw animations, but they can assemble elements together and move them around in 3 dimensions to create interesting things.

AC: For those 3D moves, you created the Move Requester. Who gave you feedback?

DAN: That was working with Jeff and Kara. I think you could do a lot fancier moves, but this was an attempt at making the program simple to use. Animation can be pretty confusing to get into, but what saves it is the preview function. At least it saves me because I can always preview something in wireframe to see if I've got it backwards before rendering.

AC: How did the move requester evolve?

DAN: As soon as I got page flipping going, one of the first things I tried to do was to move an object through perspective by hand. The perspective controls that already existed in DP lent themselves fairly naturally to doing that. I'd slide something along the ground and animate it, and it looked pretty neat, but it was a definite candidate for automation right away because it was pretty painstaking to do it one frame at a time. So that is how Move evolved.

AC: The direct overscan painting is a very useful new feature for video work. Was that a high priority on this version?

DAN: Oh yes, I put in overscan not long after DP2 shipped. I'd been wanting to do it, but it was a matter of figuring out how to trick Intuition into letting me have a full overscan screen. I finally got the inside story on how to do that. It's not that hard to do, but it's hard to find out how to do. You just type in a few lines of code, but you can never figure it out by yourself. By talking with the people at Commodore, I've been able to do it in a way which was cheating, but wouldn't become a bug later with later releases of the Workbench. That was the tricky part!

AC: Did supporting extra-halfbrite cause any special problems?

DAN: {There were} no real problems. I had to go through a lot of the code and touch it up so it would work with the 64 colors. But it was pretty simple, which was why I put it in. I think it's a pretty useful feature and it wasn't that hard, unlike HAM, which a lot of people asked for, but {which} would be really hard to put it in the way my model works. Not that HAM isn't useful; it's just too hard! So I stuck with halfbrite.

AC: You've got two more cycle registers now, for a total of 6. What the heck does anyone need 6 color cycles for?

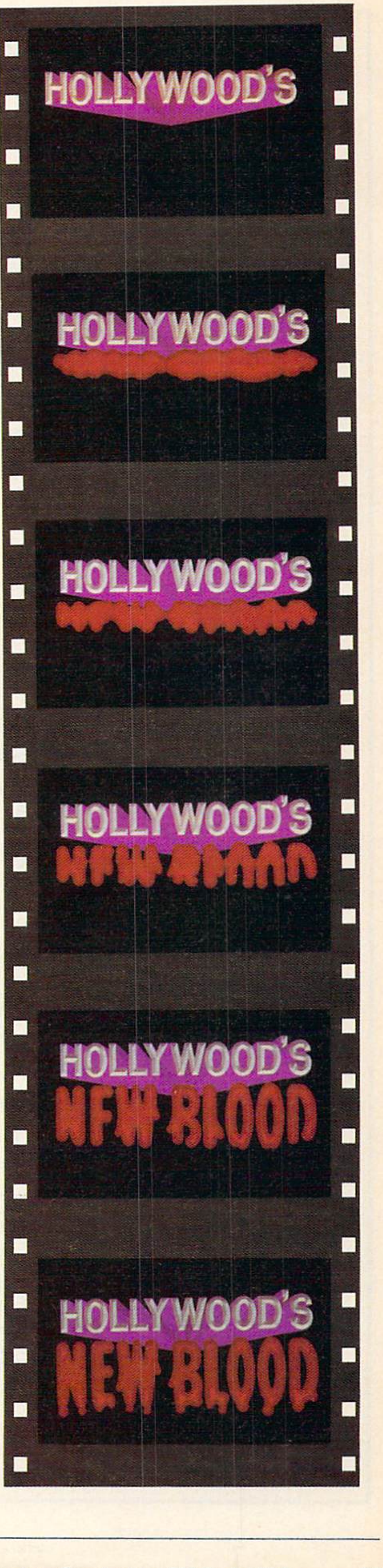
DAN: The reason is not so much for cycling colors, but for creating color shade ranges. I found that with halfbrite; I often wanted more. As a matter of fact, the default palette that you get when you boot up the halfbrite 64 color mode is set up so there's a whole lot of cycles. If you put together a range of three of the regular colors in halfbrite, you get a 6-color range. So it was mainly for shading that I put them in, not for color cycling.

AC: With that many color cycles going, you could probably crash the machine pretty easily.

DAN: In this version, I coded the color cycling code that runs on the verticle blank interrupt in assembly language, and it's much faster. So I think you can run all the cycles full speed and it still leaves enough machine going for the mouse to move around. (Note: We tried it. It does!)

AC: You've gone to 2 1/2 dimension right now. Did you consider going to a full 3-dimensional object-oriented approach?

DAN: Yes, I've thought of putting in at least a limited set of 3D objects where, instead of seeing just a flat brush, you could see a cube with sides painted to the brush. That's one of the things on my list—cubes, cylinders and things like that. I think that would be fun. Because essentially, I do have a real 3D model; I could move the brush around in 3D and easily extend that to include other objects.



Right:
Hollywood's New Blood, anim title series created by Kara Blohm

AC: It would be great to have dodecahedrons, tetrahedrons and icosohedrons flipping around the screens with little pictures on each facet. I'd go nuts with it anyway! So what is your favorite DP3 feature?

DAN: I'm happy about the way the paint tools that were already there just happened to work out to be fun to use with animation. It's more the interaction that excites me—the combinations of features that interact in different ways too numerous to understand immediately. It leaves this sort of open-endedness to it. For instance, using wrap fill with animation you can get the effect of balls with paintings on the surface rotating in space. We were just playing with that the other day, and that was something I hadn't realized before. I really don't think in terms of it as a single feature. It's more the interaction of the features that I'm pleased with.

AC: Was it scary removing the copy protection?

DAN: Yes! Some pirated versions of DP3 got out early on bulletin boards. They weren't really DP3, they were some version that was probably 2.8. Something that the pirates had renumbered as 3.0. However, sales seem to be real strong, and it could be that that actually helped us because it served as advertising. DeluxePaint III sales are mostly upgrades, but it's been way beyond predictions. It's basically flying out the door.

AC: If they talk you into doing DP4,

Quality Clips for Your Quality Art !

Acft Pics tm
Suggested Retail
$49.95

Map Pics - Worldtm
Suggested Retail $59.95

Heraldic Picstm
Suggested Retail $34.95

China Pics tm
Suggested Retail
$34.95

Christmas Pics tm
Suggested Retail $34.95

Bird Picstm
Suggested Retail
$29.95

These image-packed screens are in 16- and 32-color IFF format for use with paint packages such as Deluxe Paint II on an Amiga 500, 1000 or 2000.

All packages require AmigaDos V1.2 or V1.3, a minimum of 512K of memory and a paint package.

To order, see your
dealer or contact :
Tangent 270
PO Box 38587-A1
Denver, CO 80238
(303) 322-1262

TANGENT 270

Deluxe Paint II is a trademark of Electronic Arts; Amiga and Amiga-Dos are trademarks of Commodore Amiga.

Circle 154 on Reader Service card.

what kind of features would you like to see in it?

DAN: I've got this file full of features. A lot of them are kind of boring, like the ability to convert anims between different formats. I'd like to have multipoint interpolations, so that you could get curved trajectories. It would be nice to support all the brush transformations for anim brushes instead of just the ones that don't change the size of the anim brush. There's a lot of work here, but it's mostly just fleshing out the current model and filling in the gaps. The main limitation on DeluxePaint has been the lack of memory; suddenly, a 2-meg machine seems small. With more memory on the machine, I can go crazy, so in a way I'm waiting to see how the Amiga evolves.

AC: What's next?

DAN: I don't know. I'm going to let DP on the Amiga sit for a while. One of my fantasies is to integrate animation and music in a program that will sync actions and points in the music in a very easy way. That may be too ambitious, but it's one possibility.

Usually programmers are kept isolated from the beta testers, but Dan Silva wanted to try something different with DeluxePaint III. Instead of existing in a vacuum tied only to the real world by the occasional bug reports, Dan Silva wanted to create "a broad bandwidth of minds," working together on the program design.

Everyone wanted to be a beta tester, but very few were chosen to work in this close relationship with the legendary Dan Silva. Their influence on the design of DeluxePaint III was profound. Here is their story:

The Houses DPaint 3 Built

Jeff Bruette stuck his foot firmly in Hollywood's door with his own startup Amiga graphics house in 1987. Since then, Prism Computer Graphics has launched Amiga graphics on the national TV shows Max Headroom, Secrets and Mysteries; in feature movies like Bulletproof and Disorderlies; and for presitigious industrial clients like CocaCola, Campbell's Soup and Hills Brother's Coffee. Bruette brought Silva his typical day-to-day problems, which include delivering graphics yesterday, competing head-to-head against vicious competitors and producing the ultimate impossible effect.

An independent graphics house faces stiff odds in LA, and Jeff Bruette has relied on DeluxePaint III for almost every one of his projects since last February. He says, "If it hadn't been for DP3, I don't know where my business would be right now. DP3 has made that big a difference!"

AC: Which features in particular have bailed you out?

JEFF: Move in conjunction with the animation was invaluable. My competition had to use Video FX 3D to simulate what I was doing, which would cut them down to 16 colors. Or they could hand move the animation in DeluxePaint II, which wasn't as smooth. Or {they could} do Videoscape animation which is very time-consuming. My clients would come in and ask for a graphic of their logo flying in from infinity.

Other Amiga graphics houses take three or four days to enter the logo and hand-draw in all the frames. Instead, I could scan their logo in with a Sharp scanner, pick it up as an anim brush, and use the move requester to fly it off into the distance.

The whole process took 10 minutes, so I could play it back for the client right then and there. It was a real selling feature, particularly when a lot of my clients were Amiga users. Simply because I had the beta version, they had to use me. There was no where else to go.

The other feature that has made the big difference for me is halfbrite. I actually don't use it much for painting. Its real value for me is converting pictures. I'll scan something in with my Sharp scanner with its 24 bitplanes, which then gets converted to HAM mode. When you convert HAM images down to 32 colors with no halfbrite, they're just torn apart, so I use the Digi-View software to convert the scanner's HAM image to halfbrite. Now the image has 32 colors plus the 32 halfbrites and the dithering capabilities of Digi-View. It gives me a real beautiful picture! If you put the HAM picture side by side to the halfbrite 320 x 400 picture, you almost can't tell the difference.

AC: You've achieved some very professional results. What can the average home user do with a tool like DPaint III?

JEFF: We're putting together a DP3 how-to video with Electronic Arts. It will cover all the new features like animation, extra halfbrite, wrap brush and tint fill modes, the fill freehand, the outline shapes, and the new font requester. Our aim is to show people how to combine these features to create actual graphics, because a lot of these things are interesting standing on their own, but combining them gives inspiration to new ideas. We're covering how to do special effects like starfields, streaming stars, flying titles with shadows, and scrolling text for end credits on a video. With those combinations of effects, the average user can easily create a very hi-tech look—like Star Trek in miniature.

Pen and pencil were Kara Blohm's tools of the trade for 15 years, and she wielded them artfully for top LA graphics houses like Saul Bass and for the 23rd Olympiad, but the power of the Amiga mouse launched Kara into her own computer graphics house. Kara now earns her living creating Amiga graphics for Hollywood video production houses and selling Karafonts, two of which are included with DeluxePaint III.

Hollywood has a stringent eye for quality, so Kara works mainly in hi-res and her input encouraged Dan Silva to make DP3 capable of effects to please the most demanding clients.

AC: What do the Hollywood post-production houses think of your DeluxePaint III graphics?

KARA: I surprised one of the editors who was helping me transfer my graphics to Betacam in a video bay. She wanted to know what graphics computer I used, and when I told her it was an Amiga, she exclaimed, "That looks as good as our Quantel Paintbox!" The Amiga is wonderful that way. You can really simulate the high-end systems.

DPAINT III

(PLUS MOVIESETTER USERS)

ANIMATED FONTS

Bring your screens to life with

3D FONT--A full rotation 3D font

For effects that will knock their socks off !!

DISSOLVE FONT--Yes it does !

Dissolve on or off screen - Rotate, Shrink, etc.

POUR FONT-Pour in place **WOW**

Animated paint can pours the font on screen !

COMIC FONT-See to believe !

Animated characters that bring your title to life

OVER 270 ANIMATED BRUSHES

Thousands of screens that bring out your best

ONLY $ 39.95 Delivered to your door !

Check or M.O. to : **ANIVISION**

Two Disk SET

P.O. Box 801
PROSSER, WA 99350
WA RES ADD $3.12 TX

COMING SOON-**PREHISTORIC AND SCI - FI**

products named are trademarks of there respective co.

Circle 150 on Reader Service card.

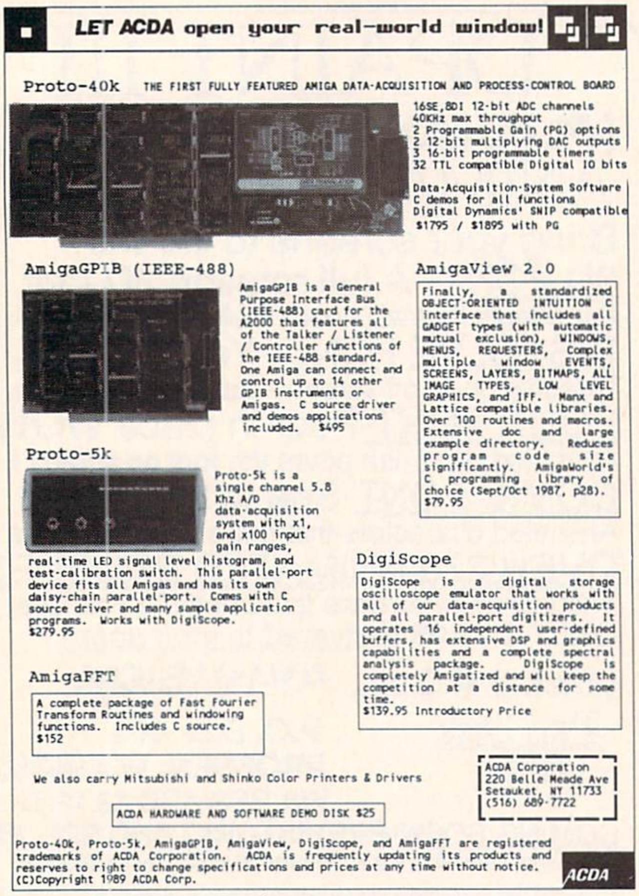
A lot of the DeluxePaint tools, like stencils and gradient fills, are similar to the Paintbox. On the Amiga, I don't get the same resolution or the millions of colors, but son of a gun, with DP3 I sure have a lot of the same tools!

AC: Has DP3 affected your graphic design style?

KARA: You could do animations before with a page-flipping program, but that's such a slow process. Now you can see how it's going to work right off the bat. DP3 makes life so much easier. I did an effect for a low-budget horror film called "Hollywood's New Blood" that had blood dripping down the screen to form the letters. I couldn't have done it without DP3. It would have taken too long.

My aim is to simulate broadcast TV effects on the Amiga, effects like little trails and glints going across logos. At first Dan just had the expanded page-flipping mode which didn't let me do fast enough glints across the letters in hi-res, but sure enough he got the compressed animation mode running, and it opened whole new worlds of what we can do with DP3 for higher level industrial graphics. Now I can do star glints and little highlights across letters which simulate broadcast TV a lot more.

AC: Since you work in hi-res, what do you think about the new fatter Agnus CHIP with 1 meg of CHIP RAM?


LET ACDA open your real-world window!

Proto-40k THE FIRST FULLY FEATURED AMIGA DATA-ACQUISITION AND PROCESS-CONTROL BOARD

16SE,8DI 12-bit ADC channels
40KHz max throughput
2 Programmable Gain (PG) options
2 12-bit multiplying DAC outputs
3 16-bit programmable timers
32 TTL compatible Digital IO bits

Data-Acquisition-System Software
C demos for all functions
Digital Dynamics' SNIP compatible
$1795 / $1895 with PG

AmigaGPIB (IEEE-488)

AmigaGPIB is a General Purpose Interface Bus (IEEE-488) card for the A2000 that features all of the Talker / Listener / Controller functions of the IEEE-488 standard. One Amiga can connect and control up to 14 other GPIB instruments or Amigas. C source driver and demos applications included. $495

AmigaView 2.0

Finally, a standardized OBJECT-ORIENTED INTUITION C interface that includes all GADGET types (with automatic mutual exclusion), WINDOWS, MENUS, REQUESTERS, Complex multiple window EVENTS, SCREENS, LAYERS, BITMAPS, ALL IMAGE TYPES, LOW LEVEL GRAPHICS, and IFF. Manx and Lattice compatible libraries. Over 100 routines and macros. Extensive doc and large example directory. Reduces program code size significantly. AmigaWorld's C programming library of choice (Sept/Oct 1987, p28). $79.95

Proto-5k

Proto-5k is a single channel 5.8 Khz A/D data-acquisition system with x1, and x100 input gain ranges, real-time LED signal level histogram, and test-calibration switch. This parallel-port device fits all Amigas and has its own daisy-chain parallel-port. Comes with C source driver and many sample application programs. Works with DigiScope. $279.95

DigiScope

DigiScope is a digital storage oscilloscope emulator that works with all of our data-acquisition products and all parallel-port digitizers. It operates 16 independent user-defined buffers, has extensive DSP and graphics capabilities and a complete spectral analysis package. DigiScope is completely Amigatized and will keep the competition at a distance for some time.
$139.95 Introductory Price

AmigaFFT

A complete package of Fast Fourier Transform Routines and windowing functions. Includes C source.
$152

We also carry Mitsubishi and Shinko Color Printers & Drivers

ACDA Corporation
220 Belle Meade Ave
Setauket, NY 11733
(516) 689-7722

ACDA HARDWARE AND SOFTWARE DEMO DISK $25

Proto-40k, Proto-5k, AmigaGPIB, AmigaView, DigiScope, and AmigaFFT are registered trademarks of ACDA Corporation. ACDA is frequently updating its products and reserves to right to change specifications and prices at any time without notice.
(C)Copyright 1989 ACDA Corp.

ACDA

Circle 104 on Reader Service card.

KARA: The fatter Agnus will give me the power to work with 16 colors in hi-res overscan and to grab big brushes like a whole logo. Right now, I can only have 8 colors in hi-res overscan, and I often don't have enough CHIP RAM left to grab a whole logo as a brush. When I get a fatter Agnus, it will really open up some doors. And wouldn't you know it, Jeff Bruette was one of the first people to get one. He taunts me with it. Everything you can think of he's got on his machine: 9 megs, fatter Agnus, Sharp scanner, everything. (Sigh)

AC: How often did you talk to Dan while you were betatesting?

KARA: Probably 4 or 5 times a week. I ran up three or four hundred dollars in telephone bills while testing DP3. I didn't mind though. It was such a joy to have it in my hands and be able to do a lot more of the effects I wanted to do.

AC: Did you have a lot of interaction with the other beta testers?

KARA: Yes, Jeff and I would call each other a lot. "Did you get it? Did you see this new thing? Oh yeah! Wow!" We had a good interaction that was very valuable. When Dan came down here to meet with all the LA beta testers. We put on this IBM program and showed Dan the airbrush, so he revised the airbrush to be center weighted, just like a real airbrush.

Later Jeff and I were talking, and he said, "Wouldn't it be nice if the airbrush worked with all of the tools?" And I said, 'Yeah, I know it would be wonderful.' I felt like we were manipulating Dan because Jeff and I would both give him the same input (Laughs). But we just wanted to impress him with how important this was. Jeff would tell Dan about it, and I'd call Dan and say "Oh Dan, wouldn't it be nice to have an airbrush that works with all these tools?" He didn't think it would be possible, but sure enough he figured out how to do it.

He was just amazing that way. He would say, "I don't know; it might take a lot of rewriting, but I'll put it on my list." And my heart would drop, and I'd say, 'Ahh...okay.' Then he'd turn around and do it! It was wonderful. He's fabulous; he's a joy to work with. He's such a nice guy.

With the input of his intrepid beta testers, Dan Silva has opened new vistas for Amiga artists and animators. Next month, we'll give you another peek behind the scenes at how the animation demos that come with DeluxePaint III were created by the nationally televised Cris Palomino, Mr. Suave-Kevin Sullivan, and Amazing ex-Disney animator Heidi Turnipseed.

•AC•

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(800)245-8525;
in CA, (800)562-1112
DeluxePaint III, $149.00
Inquiry #267

Attention Readers!

AC WANTS YOU!

Since we at Amazing Computing™ cannot determine the dependability of our advertisers from their ads alone, we want your feedback. If you have had a problem with an advertiser in AC™, let us know! Send a complete description of your exchange(s) with the advertiser—along with the names of the individuals involved—and we'll do our best to get to the bottom of things.

Ad Complaints
PiM Publications, Inc.
Amazing Computing
P.O. Box 869
Fall River, MA 02722

Scene Generator

Freely Redistributable Graphics Enhancement

review by R. Shamms Mortier, PhD.

I know, it's out of the ordinary to review a freely redistributable disk. After all, if it were truly a commercially viable product, wouldn't the author take the steps necessary to market it and make a bundle of money? Well, Scene Generator, a soon-to-be-released freely redistributable program, will leave you breathless. But first, a word about freely redistributable products.

Freely redistributable usually means free to the user. The user can also copy (distribute) the original for others. The author of Scene Generator, Brett Casebolt, actually makes a plea that this be done with Scene Generator. Freely redistributable is not shareware, where the author asks for a minute contribution, it is freeware. But I'm going to bet my booty that Version 1.1 will be marketed for sale.

After using this wonderful graphics tool for about seven hours, I called Brett and chastised him for not marketing the product right away. His response humbled me. He said he was so thankful for the volumes of shareware he has had the good fortune to use, he simply wanted to pay back the community. How's that for cold, hard, American business sense?

As it now stands, the disk could easily fetch $30.00 in the competitive Amiga graphics marketplace, and it really has no competitors. Right now, it addresses only lo-res without overscan. However, plans are already in the works to upgrade Scene Generator to other resolutions, and to include overscan as well as other features I'll mention later. Before I get into the nitty-gritty, though, allow me to give you a brief bio of Mr. Casebolt.

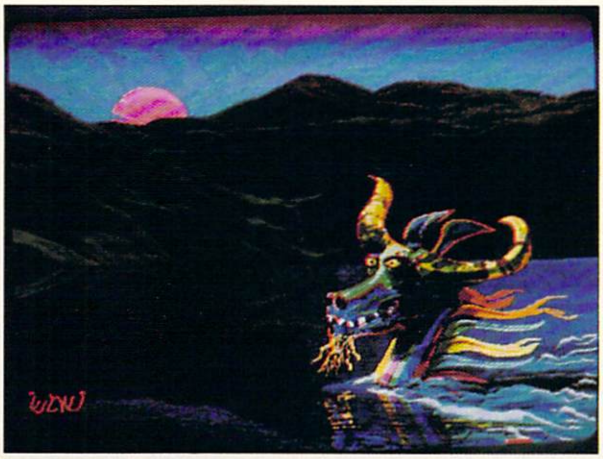
It's always nice to know a little about the history of these backyard geniuses. I guess we almost crave it, keeping in mind the sacred garages of the "two Steves" (Wozniak and Jobs, the Apple Wonderkinds). Brett is a thirty-one year-old systems programmer who works for Systems Integrators in California. A Sierra backpacker in his spare time, his job for the last four and a half years has focused on writing and maintaining assembly code for the M68000 (also the Amiga's central processor chip).

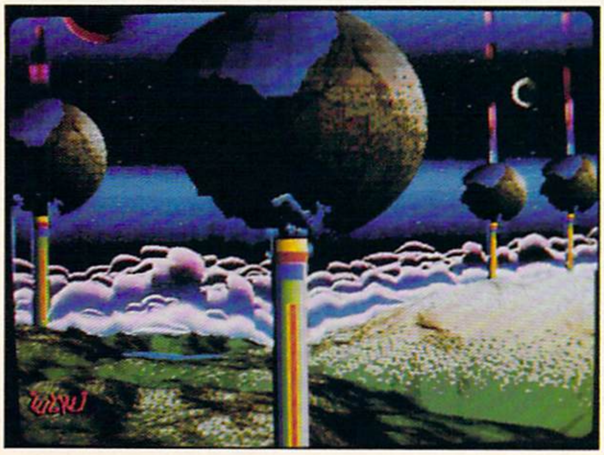
He has had his Amiga since 1983. His favorite Amiga work involves taking CPU-intensive algorithms and translating them to assembly code. It is this "hobby" that shows up so beautifully in his Scene Generator, with rendering time being reduced to a bare minimum. He originally sent the program to Fred Fish in January of 1988, and later had it published in JumpDisk (a well respected Amiga disk magazine). When he does take Scene Generator to market, he will join the honored ranks of other developers who have followed this same path from freely distributable to marketable product.

As an Amiga artist, I find this program essential for some of my recent work (exemplified by the electronic paintings that accompany this article). With Scene Generator, Brett has set yet another new directional path that others will surely expand and comment upon. Brett originally got the idea from another freely redistributable program which generates wire frames of fractal landscapes. These types of programs have received considerable press over the last few years, largely as a result of the related work being done by such notables as Lucas Films.

By following the mathematical input of the artist, these programs are able to generate very believeable and natural

Top: *"Dragon"* ***Bottom:*** *"Worldzend"*





scenes. Impulse Inc. in Minneapolis makes a 3D scene generator for their Turbo Silver package called Terrain, but the feel and use of it is very different from Mr. Casebolt's gem. For one thing, Brett's Scene Generator is very randomized, so you cannot really tell what you are going to get until you actually see it. It works so fast, though, you will actually enjoy the mystery. The Impulse program is meant for an entirely different purpose (3D animation), and it can take hours to render a design. The Terrain program, to be honest, also offers many other parameters (texture mapping, reflectance/refractance, etc.), so the two are not really comparable.

As it now stands, Scene Generator has a full Intuition interface, and you can multitask with Workbench or anything else you have the memory to run. Also, because this program will not remain in the "free" category for long, working with it now gives the user/artist community a chance to provide Mr. Casebolt with some feedback before a righteous version hits the streets.

There are five easy-to-understand pull-down menus, including the Height Menu and the Project Menu. The Height Menu has six options: 800, 1000, 2000, 4000, 8000 and Other, which allows you to set the height of your mountains from 100 to 9999. (It is best, in my experience, to stay between about 500 and 8000, but let your experimentation guide you.) You can light your scenes from over your right or left shoulder, from the front with no shadows, from directly overhead, or via backlighting. You can view the finished work from low, normal, or high vantage points. Also, you can add a dark blue pool of water at low, medium, and high levels.

The scene generation is turned on in The Project Menu. The Title can be hidden, the scene redrawn with new options, and the process can be stopped midstream (in case your favorite lines are being randomized out). The finished work can be saved on your chosen path. I have saved about sixty scenes so far, with twenty to a disk.

Brett hopes to add overscan and other resolutions in the near future (I'm sure your rapid response would encourage this). He has also been researching SIGGRAPH publications to add textured water surfaces, clouds, and trees. Plans are also underway to add a batch mode of operation, making it possible to enter parameters from the CLI, so you can generate multiple scenes and save them to disk for 2D-animation purposes.

Wish list

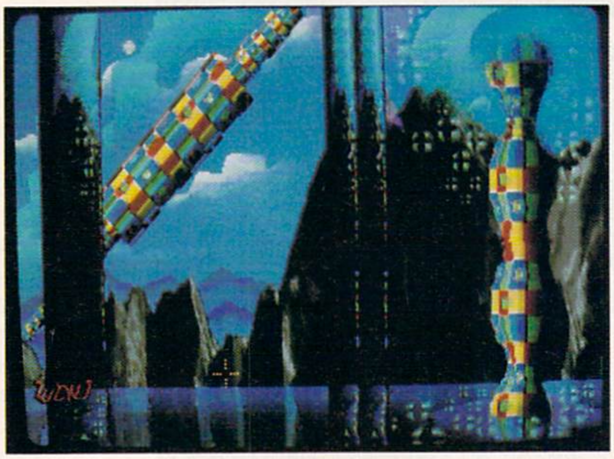
Naturally, I have my own fantasies. For one, I'd like to see an option that would utilize accelerator boards (68020/68881). Amigans are all speed junkies. Even in non-overscan lo-res, an entire scene takes about two plus minutes to render.

A future version might also compute the snow lines of certain altitudes. There should be a setting, perhaps a slider, that allows for various rough (the Rockies) and smooth (the Appalachians) renderings, perhaps by nothing more than a dithered effect. For those of us requiring fuchsia backdrops for some interplanetary scene, a color palette choice would be nice. Lastly, I've got an idea for his next product. It's called Caves. It would allow users to render multifaceted stalactited, stalagmited interiors. The artist never stops wanting more!

And speaking of art, the main reason I wanted to share this discovery with you was to show you some of the results I've been able to obtain. The paintings shown here were rendered with Scene Generator as a background tool, and combined in a PhotonPaint (MicroIllusions) environment. In PhotonPaint, I worked in the Video-Res mode in full overscan, meaning all the Scene Generations had to be manipulated in some fashion to fill out the space. Usually I just enlarged them as a brush, but sometimes I employed other choices. I am currently at work on a whole series of paintings of this type, and would be willing to discuss my other techniques with anyone interested. Meanwhile, get a hold of Casebolt's Scene Generator, and happy creativity!

•AC•

Brett Casebolt
4603 Slate Court
Rocklin, CA 95677
(FISH # 155; JumpDisk March 1989)
Scene Generator, Version 1.0, Absolutely Free



Left:
"River Trip"

Right:
"Scenario 2"

Fractal Presentation

by David Hiestand

The comforting silence was shattered by the sudden query of my wife, Jo.

"Are you ever going to do anything *useful* with that thing?"

My hands froze in the standard home position for Amiga users (one hand clutching the keyboard, the other clenching the mouse) while I mulled over the word "useful." I rolled it around my tongue, savoring it. "Useful," I muttered, "ummm ... yessss." I turned toward Jo, who was trying unsuccessfully to suppress a smile, and brightly remarked, "What a novel idea!"

Relative terms

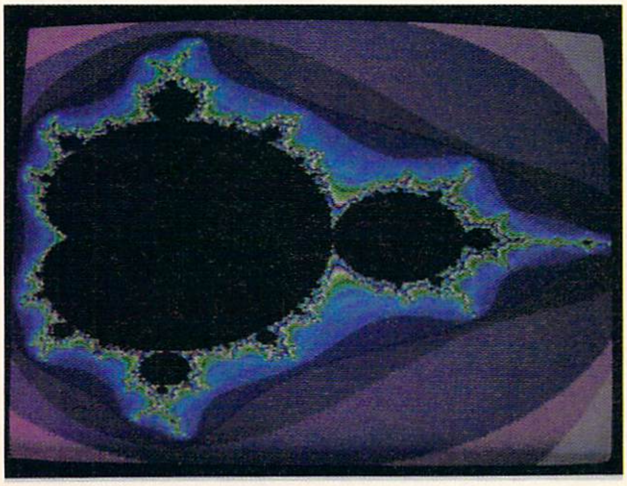
"Useful" is a relative term. Apparently, Jo and I had different perceptions about the usefulness of the "thing." I, had never questioned it. The Amiga is the principal instrument in my quest for a knowledge of fractals, which have been my obsession for the last eighteen months—or maybe more (ask my wife). But her question triggered memories of when I was chairman of the Boeing Commodore SIG (Special Interest Group), attempting to cajole, coerce, or blackmail (whichever worked) members into demonstrating to the group the many uses for their computers.

These memories, coupled with my wife's query, induced me to volunteer (a word seldom heard by SIG leaders) to speak at one of the SIG meetings and present a slide-show-like demonstration using the club's Amiga 500. By presenting dramatic pictures of fractal generations, I would be able to inform the members about fractals and their uses. Naturally, Jo would be able to sit in on the meeting and judge for herself the "usefulness" of my speech.

I totally underestimated the impact of the talk. What began as a SIG "show and tell" evolved—via numerous speaking engagements around Seattle—into a full-blown presentation, resulting in an invitation to address the Annual Northwest Math Teachers' Conference at the Seattle Center in September. I do not doubt that the relatively hot subject of fractals had something to do with this. However, much of my presentation's success must be credited to the Amiga and its software.

Carrying on the theme of "usefulness," the rest of this article will review the development of my presentation. You may be surprised at what can be accomplished with a modest array of software and hardware.

One of many fractal images presented



Having made the decision to do the original talk for the SIG, I checked my resources. (Arguably, the order of these two activities could have been reversed.) In a word, they were unimpressive. My hardware consisted of a 512K Amiga 1000 with a color monitor and an external floppy drive. I had DPaint II, some public domain and shareware software that generated fractal graphics, several of my own compiled BASIC programs that created intricate geometric fractals, and stacks of magazine articles and books explaining fractals and displaying their intriguing patterns. A copy of AEGIS VideoTitler (borrowed from a friendly SIG member) rounded out my inventory.

Perhaps my greatest deficiency was a total lack of personal experience in creating a video presentation. I did have some experience making viewchart presentations to managers, but these consisted mainly of three or four viewcharts of text, which could usually be reduced to: "We are over budget and behind schedule." These would be followed by three or four more charts saying something about why the problem existed (e.g., "The idiots on the proposal team didn't allow for enough money or time."). The last two to three charts recommended the problem's "solution," which was usually a glossed-over version of "Give us more money and time."

Picture acquisition

Early in the project I realized that there was no way to create new programs for all the fractals I wanted to show. So, without compunction, I made liberal use of public domain and shareware programs such as MandFXP V3.0 and FRACTGEN on Fred Fish 188. These programs not only had "save" features for user-generated fractals, but an IFF treasure chest of gems already generated. My compiled BASIC programs had no such "save" features, so I relied on the screen-dump program provided on the Workbench to move screens to disk.

I made another important realization during this period of "picture acquisition." I was connected to a previously overlooked, but very powerful, resource: the Amiga community, whose presence was felt in the form of pertinent public domain and shareware programs and helpful SIG members. For the fractals for which no programs existed, I petitioned one SIG member who generously donated his time and equipment to help with digitized pictures from magazines and books.

Limited RAM concerns

A half meg of RAM isn't much for processing images. In order to conserve memory for both DPaint and AEGIS, I closed all windows (saving 6 to 12 Kbytes) before loading the program. If the icon of the program is dragged out of the window, the window can be eliminated by clicking on the close gadget. The icon remains and can be double-clicked to load the program. Also, I scratched my digital clock (another 10K) and my virus checker (about 7K). In general, I deleted any extraneous background tasks to squeeze out all the extra bytes of RAM I could.

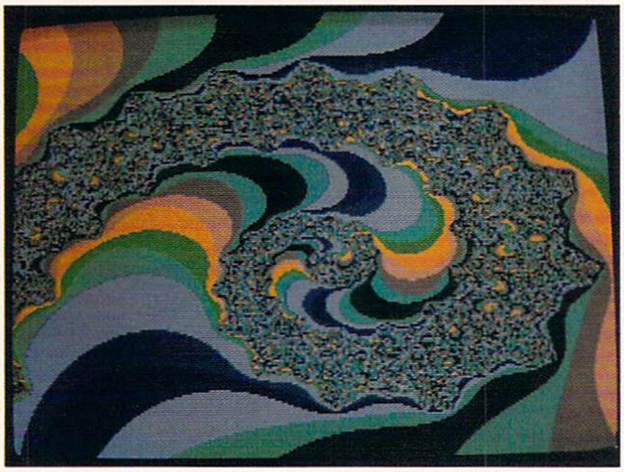
AEGIS VideoTitler

The pictures I collected represented a variety of different kinds of fractals. As a result, I needed titles to explain and separate them. AEGIS VideoTitler, which is made up of AEGIS VideoTitler and AEGIS VideoSEG, performed this function admirably. Both component programs are excellent examples of useful, user-friendly video tools.

AEGIS VideoTitler is replete with delightful features to aid in the creation of titles. A parade of fonts (diamond, ruby, topaz, garnet, sapphire, opal, brick, and others) in various pitches can be combined with a number of styles (italic, bold, underline, camouflage, emboss, 3D block, neon, outline, and others) to yield an almost endless selection of graphics lettering. When these selections are coupled with a separate color palette for background, pen, cap, and shadow, almost any conceivable lettering effect can be achieved.

As if this were not enough, AEGIS provides a separate set of four fonts called "polyfonts," which constitute a whole new approach to graphics lettering. Of course, you can import any IFF file of your choice—for example, one from DPaint—and apply any of the polyfonts to title it. Or you can stay completely within the video—as I often did—and generate colorful 3D titles with subdued backgrounds that seem suspended in space.

With limited RAM (512K) some features of VideoTitler, like "undo" and "background", are lost. AEGIS seems to have designed the program so that, as RAM increases, features are added. But even with 512K, a core of the most useful functions remain. It was a bit disconcerting when I first loaded VideoTitler. Immediately, a requester appeared with the warning, "ATTENTION!!! NOT ENOUGH ROOM FOR undo buffer. Undo buffer will be deleted." Two buttons were provided to choose from. The left button said "OK" and the right button said "OK." "Let's see," I mused, "if I click the left one, I lose. On the other hand, a right-button click will..." All I could picture was an AEGIS programmer sitting in a dimly lit room, the glow from the monitor reflecting off his face as he chortled demonically while programming this requester. I threw caution to the wind and clicked on the left "OK".



Next, I pulled down the text menu and selected Polyfonts. I then loaded the "Swan" polyfont and selected Entry. A small rectangle appeared near the top of the screen. I typed the word "FRACTALS." As I entered the letters, the rectangle stretched to accommodate them. Working the cursor to the center of the rectangle, I pressed the left mouse button and dragged the rectangle—letters and all—to the middle of the screen. So far, so good.

Next I moved the cursor to the midpoint of the top line of the rectangle, clicked the left mouse button, and dragged the line toward the top of the screen. Surprise! The bottom line of the rectangle remained fixed while the rectangle's sides stretched to follow the top line. Even more interesting was the fact that the letters, affixed to the top and bottom lines, stretched to follow this action. Dragging the bottom line toward the bottom of the screen produced a very elongated "FRACTALS."

Continuing this manner, I moved the cursor to the midpoint of the right and left sides of the rectangle, pulling "FRACTALS" to the respective edges of the screen. *Voilà!* I had created a title whose letters occupied almost the entire screen.

As an experiment I moved the cursor to the upper right-hand corner of the rectangle and dragged toward the right side of the screen. The rectangle skewed to a kite-shaped figure (parallelogram), the letters dutifully mimicking this behavior. I now had italic-like letters. While experimenting on a different title, I learned that if you click and drag the top line of the rectangle downward, passing through the bottom line and continuing, the letters formed will appear with a reflection effect. The same is true if you pull either the right or left side of the rectangle through its opposite side. Clicking the right mouse button fixes the word on the screen, makes the rectangle vanish, and allows the selection of a new font. It's interesting to note that each block (word, phrase, or sentence) created this way is treated as a separate object, meaning that one block can be laid over another. If desired, the two blocks can later be separated easily by clicking anywhere within one block and dragging it away from the other—like shifting and shuffling graphic overlay transparencies.

I really enjoyed working with polyfonts. Depending on how busy my graphics were, I was able to load three to four fonts before VideoTitler started screaming, "Memory dangerously low!"

AEGIS VideoSEG

At this point, I had pictures, titles, and explanation "slides." (In general, titling was done in AEGIS VideoTitler, but my explanations usually involved figure drawings as well. These were done with DPaint, which had the tools for this purpose.) Somehow, all my individual slides had to be put together into a coherent whole. Here is where AEGIS VideoSEG shines. VideoSEG allows you to build a script file that defines not only the order in which the slides are to be shown, but also how each is to be "wiped" from the screen to make way for the next.

The process of building a script is relatively straightforward; the program is totally menu-driven. Click on the EDIT selection and then point to a filename listed in the displayed directory. Point again to any of the variety of wipes available (dissolve, fade, flip, dribble, random, burst, checkerboard, zigzag, spiral, etc.). This information is transferred to a numbered record (called a "frame") appearing in a list in the script editor. Continuing in this manner a script file is created, each frame containing the filename and the wipe. A time delay in seconds can also be entered into the frame. This would be used if you chose autoloop while laying the script.

The editor also provides for the deletion or insertion of a record into the script list. When the script is finished it can be played using a menu selection for either manual or autoloop. I used the manual setting, since it allows the mouse buttons to control the direction in which the script is placed—left for forward and right for reverse.

The giant "FRACTALS" title I built with polyfonts is the lead-in title for my talk. It is preceded by a blank (black) screen so that the "dissolve" wipe is more effective. With a striking sparkle effect, "FRACTALS" forms slowly from a black background. As they emerge, the letters scintillate randomly—like sunlight shimmering on waves—before eventually settling into a neon-blue color. The effect is enhanced if the room lights are off and the letters are projected onto a large movie screen. On several occasions I have heard "oohs" and "aahs" coming from my audiences.

Squeezing disk bytes

AEGIS VideoSEG will handle lo-res, video-res, medium-res, hi-res, interlace, and halfbrite (as well as cycling). Much to my dismay, when I played a script containing any hi-res or medium-res files, I received a GURU visit. My limited RAM forced me to reformat all my fractals to lo-res. As it turned out, this wasn't a bad thing. Several places where I made my presentation had only RGB video projectors, the bandwidth of which is insufficient to project anything but eight-color lo-res. Lo-res works infinitely better with these projectors; most hi-res pictures become a smear of shapeless color. Lo-res has the added advantage of taking up much less disk space, allowing me to put my whole presentation onto one disk.

Fractals Defined

Fractals are patterns that repeat themselves on ever-diminishing scales. The multiple reflections of an object placed between two mirrors is an example; each "frame" is an exact replica—except in size—of the preceding one.

There are several varieties of fractals: geometric, random, and chaotic. Geometric fractals, like the mirror reflections, are exactly the same on each scale, except for size. Random fractals are slightly different on each scale, like the branches of a tree or river: the y-shaped pattern is repeated on different scales, from trunk, to branch, to twig, or from creek, to brook, to tributary, to river. Nature is replete with random fractals. Take a close look at a fern frond: it repeats the frond shape in all the branches, which are themselves repeated in their branches.

A new theory of dynamic systems called CHAOS uses fractals to explain the surprisingly complex behavior that has been discovered in systems previously thought to be quite simple. CHAOS also offers the hope of discovering that complex systems long thought to be random—like weather patterns or the stock market—have an underlying order that can be explained and exploited.

Another way to conserve disk space is to chop bitplanes. Most of my titles use only 4 colors. Why, then, carry a palette of 32 colors (5 bitplanes)? I used DPaint to reformat my lo-res titles into 2 bitplanes (4 colors) with no loss—except for the excess bytes. I tried to keep the number of colors I used for the titles and explanation slides to a minimum. At times, however, more than 4 colors were necessary. But that's O.K., because I never carried more than the necessary number of bitplanes. The saved space let me add slides as the presentation evolved. If I stumbled over a concept or forgot to mention an important idea, I added the appropriate slide(s). The slides became my notes of sorts, jogging my memory and giving the talk a smooth pace.

Pseudo-motion

Although AEGIS can handle animation, I did not attempt it, fearing that with my limited RAM I would be asking for trouble. Also, animation is notoriously wasteful of disk space, and I was smugly satisfied that I had shoehorned my entire presentation (55 IFF files) onto one disk. I had no desire to expand to more disks, even if I was able to run animations. The thought of fumbling around changing disks in the dark in the middle of my talk was not appealing.

Pseudo-motion, however, was within my reach. Much can be accomplished through the cycling of colors. I never discovered how to set the cycling in AEGIS VideoTitler, but *DPaint* can be applied to the same task. AEGIS VideoSEG has a CYCLE mode that "plays" *DPaint*'s cycling.

One slide that I created with *DPaint*, used cycling to simulate a swinging pendulum (five positions forward and five backward). Just below the pendulum was an x-y plot of the pendulum's displacement and velocity. At each successive pendulum position a corresponding point on the plot would light, indicating the pendulum's displacement and velocity. The points on the perimeter of a circle that was drawn on the screen appeared to move clockwise as the pendulum swung back and forth (very effective).

On another slide—this one of a fractal pattern—DPaint was used to select a range of color cycling. The result was somewhat psychedelic, appearing like a rainbow of different colored waves crashing onto the shores of an island.

I accidentally discovered another technique when I showed two slides of the same fractal pattern which had slightly different color palettes. The slides were displayed one after the other using a "dribble" wipe. The resulting undulat-

MIDI-Mice

Software allows you to control virtually ANY VIDEO, MUSIC, or PAINT program from your MIDI instrument

- ***Transform MIDI events into mouse events and keyboard events***
- ***Fully programmable with instant recall and edit***
- ***Multitask on any model Amiga®, MIDI interface required***

Please send your check or money order of $85.00 to (CA residents add 6% tax)

Tensor Productions
280 Mathilda Drive No. 9
Goleta, California 93117
Tel: 805-685-6245
Fax: 805-685-2994

VIDI-Mice (Visual interface) also available for $85.00
Dealer inquiries invited

Circle 141 on Reader Service card.

ing effect looked like the rippling of a wheat field in a soft breeze. Of course, the effect lasted only as long as the duration of the wipe, but it was interesting.

Showing two slides of slightly different patterns—one immediately after the other—has the effect of accentuating their differences. One of my fractal pictures shows multi-colored ivy-like leaves twining around the screen's diagonals. Outlining the edges of the leaves are reduced replicas of the same twining, leaf-like pattern. On the edges of these smaller replica leaves are even tinier leaves, and so on—as deep as I care to go.

I programmed a magnified (about 30x) view of one leaf to show this recursion. Then I decided that a piece of the magnified view, overlaid on the original (using *DPaint*), would make a more dramatic picture. An added drop shadow made the piece appear to float above the original, while arrows showed where it came from. This created a picture with an "exploded-view" effect.

I then connected the images using the two-picture technique: The original, unaltered picture was followed with a "flip" wipe, and then by the exploded-view picture. (The flip is an instantaneous switch from one screen to another.) The magnified piece seemed to leap out from the original. This effect, while not quite as startling as a yo-yo springing from a 3D movie screen, was at least emphatic.

Some pointers on pointers

I am convinced that all speakers should use a pointer to direct their audiences' attention to interesting aspects of the current slide. Pointing is no problem when the speaker is standing beside the monitor (fingers work very well), but it becomes more problematic when the slide is projected on a movie screen four feet above the speaker's head.

The cursor, of course, can make an excellent pointer in either scenario. Unfortunately, once the VideoSEG script file begins to play, the cursor—no longer needed for any menu selections—vanishes. Now, somewhere buried deep within the AEGIS VideoTitler manual, there is probably a reference that describes a keystroke that can be used to force the cursor to reappear during a script play. I never found that reference, but I did discover (somewhat unintentionally) that AEGIS Video Titler does not handle HAM IFF files (which several of my digitized fractals were). This led me to discover a way of forcing the cursor to reappear during a script play.

Multi-Forth

The Language of Innovation

If you haven't tried Multi-Forth™ you may not have yet unleashed the full power of your Amiga. This comprehensive development environment includes:

- •Local Multi-Tasking
- •Built-in Assembler
- •Turnkey Compiler (royalty free)
- •Sound Drivers
- •AmigaDos & Intuition Support
- •Complete Set of Include Files
- •CompuServe Public Forum
- •Extensive Documentation
- •PowerWindows Option
- •New price - **$99**

Call for a technical data sheet or check out our online services on CompuServe at GO FORTH.

Creative Solutions, Inc.
4701 Randolph Rd. Ste. 12
Rockville, MD 20852
301-984-0262
1-800-FORTH-OK (367-8465)
Major credit cards accepted

Circle 144 on Reader Service card.

When a HAM IFF file begins to load during a script play, a requester immediately appears stating: "ATTENTION!!! IFF Width/Height/Depth truncation." Because you have to select either the left "OK" or the right "OK" (visions of a sinister AEGIS programmer again come to mind), the cursor appears. Interestingly enough, after a selection is made and the script begins loading the next IFF file, the cursor does *not* disappear.

If nothing else can be said of me, I learn from my mistakes. To take advantage of my discovery, I created a perfectly black HAM picture, which is now the very first slide in my script. Each time it loads at the beginning of my presentation I receive an error message and—more importantly—a cursor. So much for the pointer problem. It isn't often that mistakes work in a programmer's favor, but this one does.

Spreading the word

My presentation has been well-received. If my audience is not an Amiga SIG, I try to stress that the whole presentation was made on an Amiga with Amiga commercial, public domain, and shareware software. Typically, there is a bit of good-natured fun poked at my "game" machine, especially from the IBM-clone and Mac types. But after the presentations, I usually hear thoughtful questions like, "How can I do that on my computer?" to which I respond, "You can't. Buy an Amiga." Before one presentation to an MS-DOS group, a member was asking whether fractals were some new Amiga adventure game. About a week later, I found out that the same member was telling everyone that I had used an IBM-AT for my presentation. How soon they forget...

•AC•

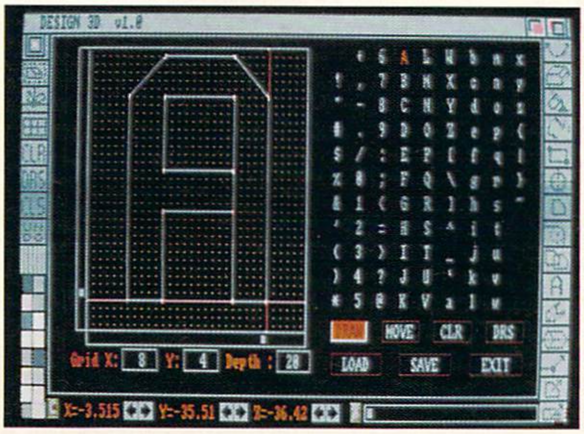
Design-3D

by R. Shamms Mortier, PhD.

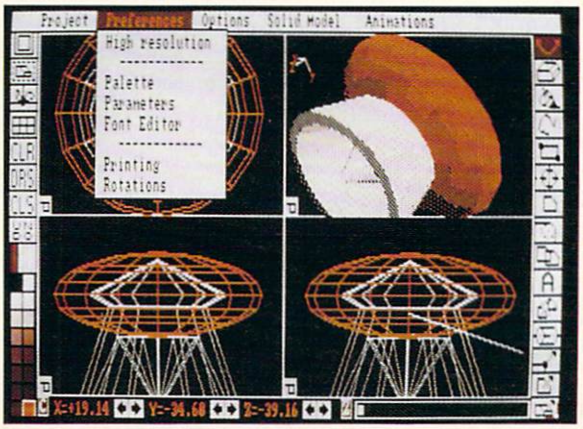
There are basically three categories into which Amiga 3D rendering/animation programs fall:

1. Top of the line serious stuff (e.g., Sculpt, TurboSilver, Caligari, PageRender3D).
2. O.K. stuff without the full functionalities of Class 1. These packages are sometimes marketed by companies to "flesh out" their Amiga product lines, or to address other companion products.
3. Bad stuff, not worth the time or money.

Class 2 releases come in a wide range of qualitative levels, from software that's just shy of meriting Class 1 status, to middling good releases that don't offer much new, to packages that are hanging on by their fingernails to keep from falling into Class 3. (Thank goodness this last category is rare in Amigaland!) Most of the Class 2 releases fall into the middle category, offering a new slant on work that other packages already accomplish, but providing little else to motivate the buyer to rush out and grab the package at all costs. Design-3D falls securely into this cubbyhole.



Redesigning the font set using the font editor.



The x, y, z, and perspective views in Design 3D.

Design-3D is a pretty package, and the accompanying manual, written in part by Nick Poliwko of PixelLight fame, is short but adequately clear in presenting what you need to get going. There are three short tutorials that allow you a modicum of discovery in Design-3D's realm of action, and the tools are defined with friendly language. The index, however, is incomplete, in that it fails to include certain key words and phrases like "ANIM" (related to the Design-3D animation format), "resolution," and "coordinates" (relevant to the finer details of object placement).

The software is not copy-protected, but you must enter a codeword from the manual to access the work screen. The opening screen comes up divided into the normative three sections, representing the X, Y and Z planes of sight. In addition, there is a fourth division that gives you a perspective view. Each of these separate views can be expanded to full screen. Only two Amiga resolutions, med-res and hi-res, are addressed (in 2, 4, 8 and 16 colors) and neither HAM nor overscan are available.

The workspace is surrounded by gadgets and menu bars which border the drawing space on all sides. On the left you find the palette requester, several "clear" tools, an UNDO gadget, and two other tools, "New Object" and "Nodes," which should be transferred to the right side with their logical neighbors. The latter, which manipulate various rendering parameters, are the actual operators: Rotate, Solid Model, Fill, Rectangle, Ellipse, Polygon, Arc, Clone, Text, Connect, Spin (the normal 3D lathe tool),

THINKER
Hypertext
for AMIGA
"..stunning capabilities..simple to operate.." "..superbly crafted.." - Gary Gehman, Amiga Sentry, 6/89

Hypertext and Outline Processing combined. Powerful Hypermedia application combines word processing and database ideas into an Idea Processor. Link applications, pictures, text. The latest technology for organizing information. Use Thinker for writing, designing, documenting, or as a database.

New Features No Credit Cards
CA res. add tax
$80 Demo Disk $5
30 day guarantee
Add $5 for COD

Poor Person Software
3721 Starr King Circle, Dept 5
Palo Alto, CA 94306
(415)-493-7234

Circle 131 on Reader Service card.

Point Select, Polygon Select, and Object Select. Some of the icons representing these attributes are not designed intuitively, so you may need to refer to the manual until they become familiar to you.

Creations can be saved out as Objects, Parts, VideoScape Objects, Screens, WireFrame drawings, and Solid drawings. A very nice feature is the ability to dump your work to a Hewlett-Packard plotter; Design-3D has a translator that addresses the HPGL language. Workbench printers are also supported. The palette requester is far from the standard that Amiga users have come to expect. It has no controls for copying colors to other color pots, no undo function, and no color cycling.

The "Parameters" requester under the Preferences section allows you the most control over various Design-3D placement functions. From here you can alter the XYZ rotation angles, the line width of the rendering grid, and the placement and brightness of four light sources. There is little instruction given in the manual on how the lights actually function, and their placement and arbitrary brightness are left up to you. This requester also allows you to set the points per circle, and the number of sectors used in the spin operation. The scale system (mm, cm, m, km, inches, and feet) can be selected, and the scale of the drawing can be set.

One 3D font comes with the package, but no information about accessing other 3D font disks is provided. A nice addition offered is the inclusion of a font editor screen, which allows you to redesign each character in the alphabet in an easy-to-understand fashion. You can also set the depth of extrusion, sculpting your 3D alphanumerics in ways that best suit your applications. For example, you can clear the letters out of the spaces altogether and replace them with your own extruded object and icon pieces.

A few miscellaneous notes: Your cursor can be either an axis or a crosshair (for finer work). "Section" directs that the first polygon entered after selecting this item will become the section of an object (only in the Solid View mode). "Copy 3D" allows the perspective window to be copied to any XYZ view.

Solid model menu

Renderings may be aborted by pressing the escape key at any time. The quality of the renderings is poor in this version of the software. By far the best is in the hi-res mode, but even then the objects take on none of the magic that so many other Amiga 3D packages promote and allow. There is a "fast" option that renders quickly, but does so somewhat sloppily. Who needs this? Under "Wireframe" you can decide whether or not you want to see the edges in a solid model or not by selecting either "Solid" or "Highlight." Under "Polygons" there are four choices: Wire (renders in wireframe mode in the palette color you choose); Solid (with no shading; the images resemble a solid color matte of the object); Solid 1 (shading using the palette colors); and Solid 2 (shading using dot patterns). Each of these choices can be used with visible or invisible wireframes.

None of these rendering options give professional or even visually suitable results, although selecting hi-res in combination with the two solid modes can allow for some interesting shading variations. The rendering module needs extensive revision to bring it up to the standards that Amiga visual artists expect and deserve. Hopefully, these inadequacies will be corrected in a future upgrade.

Animation

I have no idea what ANIM standard this package addresses, and the manual provides no hint. What I do know is that it works by a simplified scripting system that can be written with any text editor. The scripts allow for rotation around XYZ axis, zooming on the Z axis, and movement on the XY plane. If you do not select "full screen" from the options menu, the tool icons will be saved with your animations. Now, why would this even be an option? Who needs to save tool icons in an animation? Your animations can be rendered and shown in real time for preview or saved out to disk for replay, and can be rendered in wireframe or solid form.

We are no longer living in a time when Amiga users greet the release of a package like Design-3D with cries of, "Thank God! An Amiga 3D software package!" Happily, that era is behind us. There now exists a wealth of innovative, professional packages to choose from, with more visual "miracles" on the way.

When I see a company like Gold Disk producing a 3D rendering/animation package, I am inclined to purchase it, largely because their Amiga track record is excellent and established. And, although I would rather have poured accolades on the efforts of Gold Disk and an obviously talented developer, this package, in its present state, is not going to help sell any Amigas. It needs to undergo serious reworking to bring it up to par. I hope the revision process is already underway. •AC•

Design-3D
by Arnaud Ribadeau Dumas
Distributed by:
Gold Disk
PO Box 789, Streetsville,
Mississauga, Ontario
Canada L5M 2C2
Suggested Retail Price: $99.95
Inquiry #261

Tshell

A UNIX-like substitute command line interface

review by Rich Falconburg

AmigaDOS is a capable and flexible operating system that provides features not found in other personal computers. But after you become familiar with the various commands and learn some of the power they provide, you'll eventually come up against the system's weaknesses. Even with all the help given by the numerous public domain programs written to fill the gaps, you'll soon find that you may need to write a program to do seemingly simple tasks. For most Amiga owners, this means using AmigaBASIC, but because of the overhead AmigaBASIC requires, this is rarely an acceptable alternative.

Having spent much of my time writing and using command scripts in the VAX/VMS and UNIX environments, I naturally wanted the same power and flexibility on the Amiga. AmigaDOS comes so close it's frustrating. Seemingly simple tasks are impossible to do. I've been hoping that someone would come up with a suitable improvement without sacrificing some of the excellent features available in AmigaDOS. In the next few issues I'll examine a new program that gives Amiga users an alternative command environment.

Metran Technology of Tampa, Florida sells a program titled Tshell. This gem provides a substitute command line interface similar to the various shells available for UNIX. As you have learned, the CLI on the Amiga lets us use a variety of commands to communicate with the system's low-level functions. This is a limited "shell" that protects you from the grim realities of trying to communicate directly with the Amiga's hardware. The level of sophistication and methods of doing things vary from one shell to another, but each gives us access to the heart of the system.

A shell is an environment allowing the user to manipulate the machine's various resources. Those of you who are familiar with the UNIX command set will find Tshell a comfortable environment. The primary UNIX shell commands are available along with a few that are specific to the Amiga. There are some subtle differences here and there, but overall you'll find equal—and in some ways greater—power available to you. Those who have wanted better decision branching, file access, and access to the Amiga's graphics or Intuition interface without having to purchase a C compiler will be delighted to learn that Tshell "can do." Although the programming syntax used by this shell has been designed to follow closely that of the C language, I think anyone with even a little programming experience will be comfortable with it.

Tshell is an enhancement of, not a replacement for, the Amiga's environment. It allows full access to the normal AmigaDOS command set and resources, as well as its own environment, so you don't give up anything by using it. In fact, the distribution disk includes the ARP commands for you to use. Tshell redefines the normal device/volume structure to that of the UNIX equivalent. This means that ROOT is now a "real" root directory, in that each of the devices are seen as subdirectories of the single ROOT (/), rather than as individuals.

There's more to this than meets the eye. You can now move files with a single command, even between devices. Here is a list of the commands that are included:

Tshell commands

assign	break	cat	cd	chmod
clear	continue	cp	dat	df
dimmer	echo	envs	exit	file
fnote	help	kill	ld	load
ls	mem	mkdir	mv	path
pri	print	printf	ps	pwd
read	receive	return	rm	send
shift	sleep	stack	strcat	strcspn
strlen	strsub	tail	tasks	test
time	tsh	vars	wc	

Let's examine these one at a time. I'll list the equivalents to AmigaDOS commands where they apply. If you have never been exposed to UNIX, you are about to learn why many people call it a "programmer's operating system" or "user-unfriendly." Granted, some of the command names are a bit unusual, but you do get used to them in time, and perhaps even grow to prefer them—especially if you hate typing.

As with UNIX shells, Tshell commands are case sensitive (more on this later). Most of the commands listed above are built into the Tshell environment. That is, they are "in memory" and are only called in from the disk drive once before execution. From then on the command will remain resident in memory. Now before you start mumbling about that being "just fine" for someone with lots of memory, you should know that this shell was designed with the user in mind. The ability to configure the environment to our preferences is something we Amigans have come to expect. With Tshell, this ability to customize reaches a new level of sophistication. If you don't use a command it isn't loaded into memory. If you don't want commands that have been used to stay in memory, you just have to set a value in one or both system variables, TLBIN and TLSYS. On the other hand, if you have memory to spare, you can bring the entire command set into memory once with the **load** command. I'll cover this in more detail later. Now to the commands.

The **assign** command is identical to the WB 1.2 ASSIGN command. Because the shell keeps track of things separately from the operating system, the Tshell command must be used instead of the AmigaDOS command.

The **break** command is not the same as the one in AmigaDOS. Instead of killing a CLI, it is used to exit a block of statements in a shell script.

The **cat** command is the same as the TYPE command with some nifty features added. Tshell has a built-in paging capability (similar to the PD "more" command) that most Tshell commands use when displaying text. The cat command takes this a step further with the following keys that are active while the file is being read:

cat key commands

d	Down: similar to pressing RETURN, which causes the next page to be displayed. Pressing d will clear the display first, making the update faster.
h	Half: continue down for a half page.
f	Forward: skip a page and continue.
F	Forward: skip two pages and continue.
b	Backup: go back a page.
B	Backup: go back two pages.
t	Tag: remember this location.
g	Goto Tag: jump directly to the "tagged" location.
R	Rewind: return to the beginning of the file.
c	Continue: no more paging in this file.
q	Quit: exit this file (same as pressing ESC). If more than one file was specified, display the next one in the list.

If no filename is given, cat will take input from the keyboard. Full redirection is supported.

The **cd** command is similar to its AmigaDOS counterpart but closer to that found in the UNIX shells. As with UNIX, an environment variable called HOME may be set. This informs the cd command where to go if no path name is given. For example:

```
) HOME = WP:Documents/Mail
```

(The ")" is the default Tshell prompt.)

Entering

```
) cd
```

will now set the current directory to WP:Documents/Mail. This may be verified with the pwd (print working directory) command.

```
) pwd
WP:Documents/Mail
```

As mentioned earlier, the ROOT directory is the "daddy" of every directory in the system. In UNIX, this directory has the unique name of "/". If you enter

```
) cd /
```

The current directory will be the ROOT directory containing subdirectories of df0, df1, dh0, dh1, etc...

For AmigaDOS users this will be frustrating until you get used to it. UNIX users will feel right at home. Yes, the period (.) for the current directory and the double period (..) for the parent directory are supported. To move "up" one level in the directory tree you must enter

```
) cd ..
```

To move more than one level, you must provide the delimiting slash. For example, if your current directory is df0:devs/clipboards, to return to the root level of the volume in df0: you would enter

```
) cd ../..
```

The colon (:) is still considered the volume root level so you could use it instead.

The Tshell cd command adds some switches to make moving around flexible. The "-t" and "-b" switches let you save your current location and then recall it easily. For example let's say our current directory, GRAPHICS:PAINT/PICTURES/LORES/HAM, is several levels deep. For some reason we must change to a subdirectory in a different path, but we need to come right back to our present location. Normally this would mean a lot of typing, but using the following syntax we take some shortcuts:

```
) cd -t GRAPHICS:PAINT2/BRUSHES/LORES/HAM
```

This will save our current location and change the directory to that shown above. The cd command will place each path in a "last in, first out" buffer, also known as a stack. The pwd command will display this stack showing the first entry on the top and the last entry on the bottom.

```
) pwd
GRAPHICS:PAINT/PICTURES/LORES/HAM
GRAPHICS:PAINT2/BRUSHES/LORES/HAM
```

Only the paths given with the "-t" switch will be placed on the directory stack, sort of a marker saying "remember this directory." To return to the last directory saved this way, enter

```
) cd -b
```

One other switch is provided for the cd command, but I was never able to get it to work. This is the "-p" or "previous directory" switch. In theory, it is supposed to remember the last directory visited and return you to it.

And last, but not least, the system variable DEFCD determines whether or not the "implied" cd function is operative. With this option enabled, simply typing the path name will cause the current directory to be changed to that entered.

The **chmod** command emulates the PROTECT command. The real advantage here is that the cp (copy) and mv (move) commands will use all of the appropriate protection bits. The chmod command also supports wild-carding.

The **clear** command is used to unset various variables and procedure definitions. It may also be used to clear the contents of the window.

The **continue** command is used in scripts to step through loops.

The **cp** command is a powerful COPY command. One major difference here is that no status information is printed unless you request it with the "-v" switch. If the "-f" switch is not given, all protection bits are respected. The "-u" switch may be used to update files. Using it will cause cp to overwrite only those files that are newer than an existing destination file, and i you prefer a menu type of copy utility, you could specify the "-i" switch. This will cause a request to be displayed showing the file name and an arrow showing where it will be copied to. At this point you would enter one of the following:

y	Yes: copy the file
c	Continue: copy this one and all the rest
n	No: skip this file
q	Quit: don't copy any more

The cp command will attempt to copy the entire source directory structure to the destination directory, unless it is overridden with the "-t" switch. This can be troublesome, and I would prefer if the default ignored the pathing and just copied to the absolute destination.

The **dat** command is an alternate form of the date command. Its output looks like:

```
) dat
Tue 17 Jan 1989 19:24:02 pm
```

The **df** command is functionally identical to the INFO command. If no device or path is specified, the statistics of the current default disk (the one the current directory is on) will be displayed. Wildcards may be used.

The **echo** command provides a few switches that will be welcomed by many programmers out there. In addition to the "-n" switch, which suppresses the "newline" character, there is also the "-B" switch. With it you can cause numbers to be displayed as:

-Bb = Binary
-Bo = Octal
-Bd = Decimal
-Bh = Hexadecimal

The **envs** command will list all currently defined environment variables.

The **exit** command is used to quit out of a script immediately. If it is entered at the shell prompt, the shell and window closes as though you had entered ENDCLI. Unfortunately, there seems to be no way to exit a shell script and close the window at the same time. A numerical value may be included; it will be used as the return code of the program.

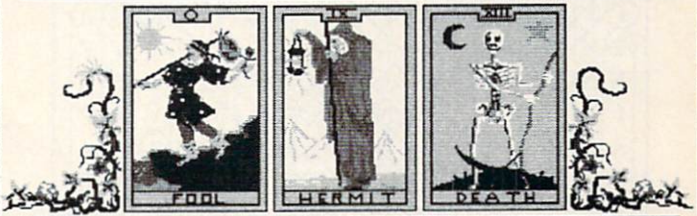
The **file** command is one of the nicest ones included. True to its UNIX counterpart, this command will read the first 100 bytes of a file and try to guess what kind of file it is. It prints a short description such like:

```
Workbench icon position file
Workbench version 1 tool icon
Workbench version 1 project icon
Workbench version 1 drawer icon
Workbench version 1 disk icon
text file
Amiga load file (executable?)
directory
```

...and more.

THE TAROT MASTER

Bring the Ancient art of the TAROT to your Amiga



The TAROT MASTER from Empire Graphics is the first fully animated fortune telling software package for the Amiga. The TAROT MASTER not only entertains you and your friends with it's authentic Tarot readings, but it will also guide you in the interpretation of the mystic Tarot.

* Two full disks of graphics and animations
* Fully animated Major Arcana
* Classic 3 & 10 card spread personal readings
* Teaches definition/interpretation of all cards
* Instructs users in proper laying of cards

Experience your future today,
Ask your retailer for
The TAROT MASTER
or mail check or money order for $29.95 to
EMPIRE GRAPHICS
P.O. Box 964
Union, NJ 07083
Send $2.00 for demo disk and coupon

Amiga is a registered trademark of Commodore-Amiga, Inc.

Circle 151 on Reader Service card

If the "-r" switch is used. The file command accepts a single file as an argument and will return a numerical code that identifies the type of file. The documentation contains a complete list including codes that identify the Manx version of the object module and indicate whether it is a Manx or Lattice library file.

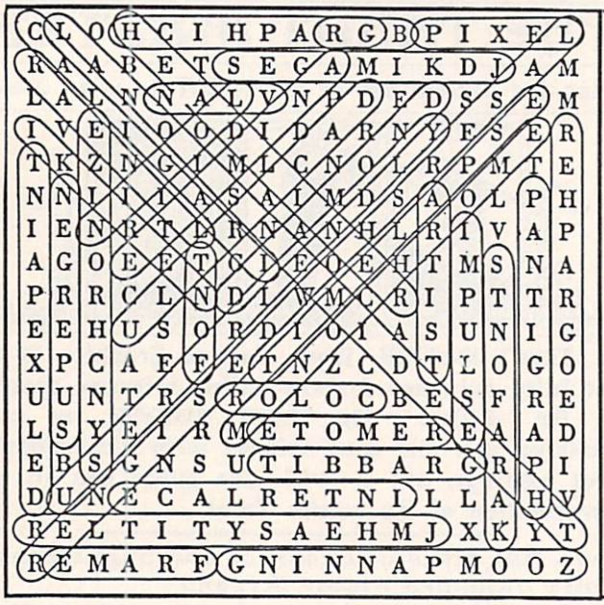
The **fnote** command is equivalent to the FILENOTE command. It's used to attach a note to the files directory entry and may be listed using the **ls** command.

The **help** command is an extremely useful and welcome addition to the Amiga command repertory. The Tshell distribution disk contains directories of text files used in conjunction with this command. There are two ways to use the help command. The easiest is to simply press the HELP key, which produces a formatted listing of the commands with Help available. The first time you press HELP, there will be a noticeable delay while the entries are read into memory. This occurs in each shell window. You then type in the listed name of the command or subject that help is needed on, and press the HELP key once more to print an abbreviated list of the command's syntax and usage. If more explanation is needed, press HELP again. The other way is to type:

```
) help "command"
```

where "command" is the command you are interested in. You can add your own help files to the directories and use the help command to recall the information.

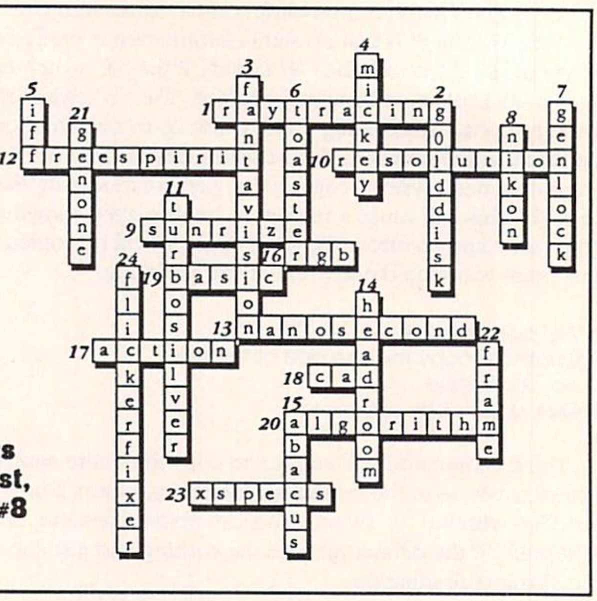
The **ld** command is used to list the contents of a directory. The design of the companion command ls makes this command



C L O H C I H P A R G B P I X E L
R A A B E T S E C A M I K D J A M
L A L N N A L V N P D E D S S E M
I V E I O O D I D A R N Y E S E R
T K Z N G I M L C N O L R P M T E
N N I I I A S A I M D S A O L P H
I E N R T L R N A N H L R I V A P
A G O E E T C I E O E H T M S N A
P R R C L N D I W M C R I P T T R
E E H U S O R D I O I A S U N I G
X P C A E F E T N Z C D T L O G O
U U N T R S R O L O C B E S F R E
L S Y E I R M E T O M E R E A A D
E B S G N S U T I B B A R G R P I
D U N E C A L R E T N I L L A H V
R E L T I T Y S A E H M J X K Y T
R E M A R F G N I N N A P M O O Z

HARD Copy

Answers for August, 1989 V.4#8



almost necessary. The only functional difference this user could find was with wild-card usage. It would be much more convenient (and common to other UNIX style shells) to be able to enter:

```
) ls df0:
df0:
```

With the Tshell implementation, this simply prints as shown above. The same thing occurs with any device or directory that is specified. To see the contents, substitute the ld command or use a wildcard as in:

```
) ls df0:*
```

There is one function of the ld command using the special wild card "@" which is similar to entering DIR opt a (or DIR ALL in 1.3). I'll discuss the wild cards, redirection, and other special operations in our next issue. The switches that may be used with ls and ld are:

-l	Long listing—similar to the output from LIST
-c	No directory highlighting
-p	Include the complete path in the listing
-e	(ls only) Used to print an error message (useful in scripts)

The **mem** command will display the amount of memory available and allocated. Information for CHIP and FAST memory are shown.

The **mkdir** is used to create new directories and is similar to the AmigaDOS MAKEDIR command. As with most of the commands shown, multiple names may be entered following the command. If any of the specified directories already exist, or the name given is illegal (as defined by AmigaDOS), mkdir will complain.

The **mv** command performs most of the functions of the RENAME command but gives us more flexibility in moving and renaming files and directories. As you may have guessed, mv is short for "move," and it does this operation very well. You may even move a file or group of files between devices. Several optional switches influence the operations being performed. These include:

mv parameters

-f	Force write: using this switch will cause mv to ignore the protection bits.
-v	Verbose: tells what is going on. Without this switch no output about the operation is displayed.
-i	Interactive: waits for you to acknowledge the operation by entering one of the following:
y	Yes
n	No
c	Continue (no more prompts)
q	Quit
-u	Move the file only if the source file is newer than the destination file (if it exists).

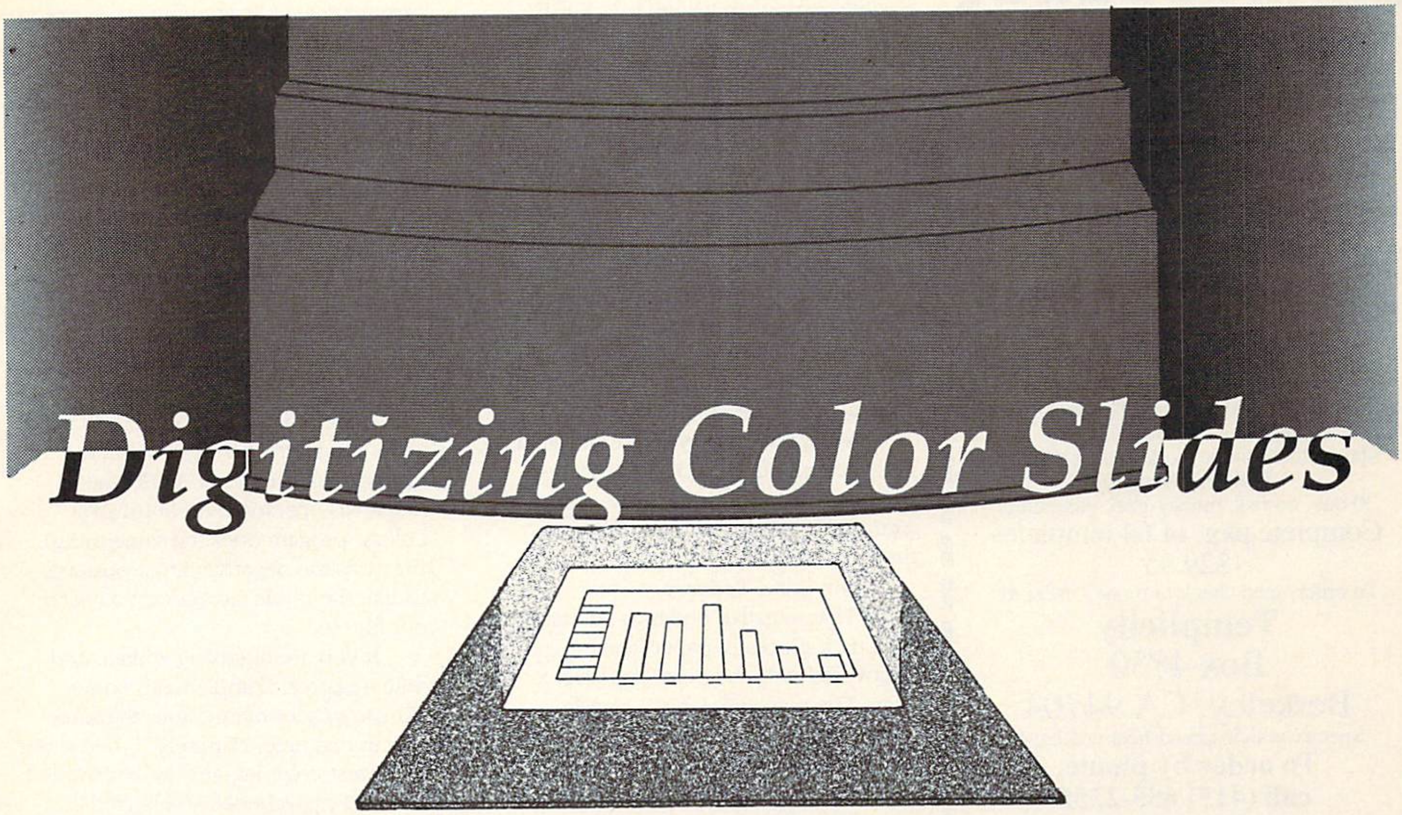
When a file is moved from one device to another, mv will first copy it to the destination and then delete the source file. This is an important distinction of the "move" command which is easy to forget.

It seems I've run out of space this issue. Next time I'll cover the rest of the commands, and we'll look at some of the unique features that Tshell provides to the command line environment. Tshell may be purchased from:

Metran Technology
P.O. Box 890
West Oneonta, NY 13861
(607) 432-4836
Inquiry # 268

•AC•

Digitizing Color Slides



by Ron S. Gull

Digitizers are a handy way to squeeze real world pictures into the Amiga to be handled by the powerful image-manipulation programs. However, as anyone who has spent any time at all behind the color wheels can tell you, it's not always just a point-and-click operation. And sometimes, it's a case of WYSANWYG (What You See Ain't Necessarily What You Get).

One possible area of difficulty involves lighting and camera placement. Even with the currently available copystand/lightstand setups providing "optimum" placement, and hours of careful setting up, the dreaded "halo effect" can still rear its less-than-lovely head, especially when digitizing to 32-colors. Also, consumer color video cameras do not provide the best available resolution and, let's face it, most of us who already own a color camera are usually less than eager to shell out at least two more C-notes to cover the cost of a dedicated black-and-white camera.

Macro digitizing

Enter macro digitizing. Most consumer cameras provide a "macro" setting on the lens which allows the camera to take a picture of something that is almost touching the lens. Most people do not make much use of this setting simply because it's very difficult to get light on a subject without seriously backlighting it. But backlighting is the very key to good digitizing with slides and negatives.

There are several benefits to macro digitizing. First, you'll increase the perceived resolution. (In my case at least, there is *definitely* less haloing in my macro files than in my conventionally produced ones.) Second, when you digitize from negatives instead of finished photos, you save a generation of image degradation. Finally, you'll likely receive a lower electric bill at the end of the month because you'll no longer be running two bright, hot lamps.

We are, of course, assuming you have a digitizer. I use SunRize Industries' Perfect Vision system. The principles explained here will work with Digi-View by NewTek as well. I simply bought Perfect Vision because of its ability to pull pictures from moving video, but (possibly) sacrificing some sharpness Digi-View would have provided.

When digitizing from negatives, one additional software requirement is needed to convert the negative images to positives. A program called PIXmate, from Progressive Peripherals, does the job for me (another program, Electronic Arts' DeluxePhotoLab, will also do the trick with its "Colors" program).

Setting up

The first step in successful macro digitizing is to get complete and even backlighting. All this requires is a light table. Normally used by photographers to view transparencies and negatives, light tables are available at any photography supply store.

If you favor "rolling your own" (like I do), you can easily build your own light table using a sheet of translucent white Plexiglas, a florescent lamp (these can be bought for as little as $10), some type of frame (I built mine into an old-but-sturdy table), and a pane of glass. One lamp should suffice for 35mm slides and negatives; however, more than one may be necessary if you intend to digitize from larger formats.

The glass is useful for two reasons. First, it holds the negative flat (more on that shortly). It also provides a smooth surface for the Red-Green-Blue filters to move between the target and the lens. You'll need this, because there will be just enough room to get the filters in.

A flat negative is an absolute necessity for macro digitizing. When a

Templicity™

64 Ready-made Templates for Your Amiga Spreadsheet!

Available for Analyze!, MaxiPlan, Superplan, and VIP Professional

Personal Financial Planning
Real Estate
Federal Income Taxes
Loan Analysis and Amortization
Home and Business Budgets
Accounting and Bookkeeping
Many other unique worksheets included!
Full User Support
30 day, no risk, money-back guarantee!

Complete pkg. of 64 templates
$29.95

To order, send check or money order to:

Templicity
Box 4850
Berkeley, CA 94704

Specify which spreadsheet you have.

To order by phone, call (415) 655-2355

Mastercard and Visa accepted.
CA res. add 7% ($2.10) sales tax.
(Offer good till 12/31/89)

Circle 157 on Reader Service card.

lens is set to "macro", the depth-of-field (the thickness of the in-focus portion of the camera's view) is very shallow. Even a slightly curled negative can mean that as much as half the frame will be badly out of focus. This curling can also present a problem when attempting to digitize color slides, particularly older heat-mounted slides.

Focusing the relative placement of the camera to the light table is also important. They should be as perpendicular to one another as possible. I go to the extreme of using a small torpedo-style level to assure everything is in line. I'm also assisted by the fact that my camera (an RCA CKC 020) has a flat back parallel to the focal plane. Most other cameras have some kind of contoured case. Just keep trying until you get it straight. A little time spent on this will definitely pay off.

One more factor in the basic setup is to get everything as mechanically stable as you can. By this I mean making sure the table holding your light table doesn't wobble. Also, be sure your tripod is either locked down tight or at least positioned so the filters can be slid between the lens and subject without touching the camera itself once it's focused. You should also consider the type of foundation your working area is built upon. If your house or apartment has wooden floors, or if you live in a mobile home, place your camera setup as close to a corner of the room as possible, so as to minimize the vibration of, say, someone walking across the floor. When you're working this tight, every little move is magnified, and can spoil a good shot.

Well, now that we have our light table set up, our tripod locked down tight, our negative under glass, and the Amiga is humming away happily, it's time to shoot, right?

Not quite. Patience.

That negative needs to be masked with dark pieces of paper, leaving uncovered only the frame we wish to snap. The reason for this is simple. Consumer video cameras usually have an "auto-iris", which is an electronically-set aperture setting. An auto-iris can be fooled into thinking the negative is just a heavily backlit person. Thus, it will "close down" to prevent overexposure. You can get a shot this way, however it will always appear washed-out in the final product. It's better to take another 30 seconds and mask the frame. If you are digitizing a mounted slide, the slide carrier is usually masking enough.

One final step before shooting is to briefly swap your composite video cables to your monitor to show what your camera is sending into the computer. Don't use Y-adapters to split the signal for the luxury of not having to swap cables. Splitting this way will degrade the video signal noticeably. My wallet dictates I use a Commodore 1702 monitor (a hangover from my C64 days—a Mimetics AmiGen genlock provides composite video output). I keep the monitor's tint, color, brightness, and contrast controls set at their detents, and set the camera's white balance so the output most closely matches the actual color of the target. So far I've been very lucky. I've put my digitized IFF files on a friend's Amiga with a Commodore 1084 monitor, and things are usually very close.

OK, we're finally going to shoot. Dial in your camera's "macro" setting and focus on the target through your Green filter. Due to the fact that the lenses of consumer cameras are not usually color-corrected, there will usually be a noticeable difference in the focus when swapping between the filters. Focusing through Green is the compromise that seems to work best.

Shoot!

Go through your filters, hit the Draw button, and voila!—a digitized negative (hopefully a reasonable facsimile of the one you've got on your light table). Congratulations.

"OK", you ask, "now what do I do with the negative?"

Once a file is saved, go to PIXmate. In the Image Processor, you'll find a logic operation entitled "EOR"(the "Negative" operation in PhotoLab's "Colors" progam does the same thing). This turns the negative into a positive, just like the photo processor you send your film to.

If you are digitizing a black-and-white negative, Complement, under PIXmate's Color menu, does the same thing in real time. Naturally, I tried it out on my first color image. I was rewarded with a very psychedelic,"solarized"-looking photo!—very handy for special effects!) PIXmate's Contrast Control, on the "Color Bias" panel under the "Color Menu", can also help tweak the images for better highlights. Experiment!

Well, there you have it—a basic guide to macro digitizing. I've found it to be a very useful addition to my graphic Bag o' Tricks.

•AC•

SunRize Industries
3801 Old College Rd
Bryan, TX 77801
(409) 846-1311
Perfect Vision, $219.95
Inquiry #262

Progressive Peripherals & Software
464 Kalamath St.
Denver, CO 80204
(303) 825-4144
PIXmate, $69.95
Inquiry #263

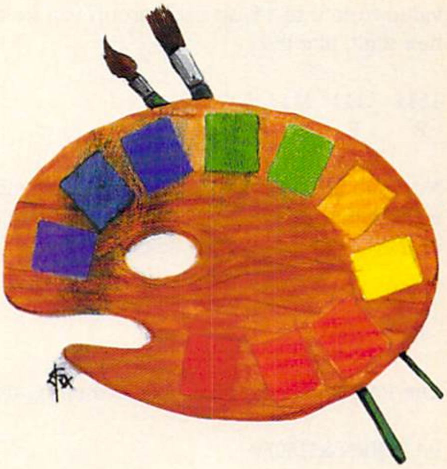
NewTek
115 W. Crane St.
Topeka, KS 66603
(913) 354-1146
Digi-View, $199.95
Inquiry #264

Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
(800)245-8525; in CA, (800) 562-1112
DeluxePhotoLab, $149.95
Inquiry #265

High Octane Colors

Maximizing Your Colors in AmigaBASIC

by Robert D'Asto



One of the first things AmigaBASIC programmers learn is how to add colors to their programs through the use of screen depth. If more colors are desired, one simply increases the number specifying the bitplane parameter in the SCREEN statement. One to five bitplanes gives us two to thirty-two flavors to choose from, and that's all there is to that, right?

Oh, I almost forgot—for the more adventurous souls, there's extra-half-brite and HAM modes which can produce 64 or 4,096 colors respectively. Now that's *absolutely* all the color programming variations possible on the Amiga, right?

Not necessarily. Just as a painter can mix two or more pigments together to produce a new and different hue, so can the programmer "mix" pixels on the Amiga to substantially expand his palette choices. This is done with a technique called dithering.

Dithering is the process of combining pixels of different colors on the screen to create another apparent hue. It's a sort of optical illusion. As any color photographer knows, if one were to mix red light with blue light, the result would be magenta—a combination of these two colors. Likewise, mixing red and green produces yellow, a green-blue combination gives us cyan, and so on. We cannot actually combine colors on a computer screen, as every pixel must be of a single color at any one time. We can, however, "mix" pixels of different colors in close proximity on the screen to create the same effect.

For example, a tight grouping of pixels which are alternately set to the colors red and blue will appear to be a single patch of magenta pixels. The effect is quite convincing, especially with higher resolution modes where the individual pixels are smaller and more densely "packed."

Let's review a few fundamentals and see how this can be worked out from a programming standpoint. Imagine a repeating pattern of pixels which are alternately red and "no color" (the background color), like this:

```
1010101010101010
0101010101010101
1010101010101010
0101010101010101
...etc
```

Each digit represents a single pixel. The digit 1 represents the red pixel. The digit 0 represents the pixel pertaining to the background color, which is blue in the default palette. Notice that the lines are "staggered" so each 1 has a 0 on each side, as well as above and below it. When viewed, a pixel arrangement like this would appear magenta—a combination of red and the blue background. This sort of setup is easy to program with the PATTERN statement and will produce a dithering effect all by itself. I will review the use of this keyword in a moment.

But first type in Listing One at the end of this article and run it. You will see three white rectangles, one filled with red, the other with blue, and the third with a shade of magenta, all with a black background. If you count all the colors now visible on the screen you should have five—black, white, red, blue and magenta. That's five different colors on a 2-bitplane screen, which is supposedly only capable of 4 colors. The fifth color, magenta, is a mixture of the red with the blue background, and is known as an "apparent" color.

As you can see, it is a convincing effect, especially since we are using the default hi-res (640 x 200) screen. The smaller the pixels, the better the colors seem to blend together. Try this same code on a lo-res (320 x 200) and an interlaced (640 x 400) screen to see how pixel size affects the illusion.

The bit pattern is created using the PATTERN keyword. For those unfamiliar with the use of this statement, I will give a quick description. The PATTERN keyword is used to create a "bitmask" for all graphics rendering including lines, text, and area fills. A bitmask is like a stencil. If you cut some holes in a piece of paper and lay the paper down on a surface, you can then apply paint to the paper and produce a pattern on the surface which matches the holes. A bitmask works the same way, but with pixel colors instead of paint.

There are two applications for the PATTERN statement: line patterns or area patterns. To create a line pattern, the syntax is

```
PATTERN linepattern%
```

where linepattern% is a short integer value defining a 16-bit bitmask, which will act as a stencil for all lines drawn. You will have to think in terms of binary numbers to determine this value. A solid line would have a bitmask which looks like this:

```
1111111111111111
```

This is a 16-bit binary number with all bits set to 1, which is equal to a large decimal number. The AmigaBASIC interpreter will not read binary numbers directly, so we have to convert the value into a numbering system it can read. We could convert it to decimal, but that's a time-consuming and error-prone method. The hexadecimal system is much easier. This system counts from 0 to 15 like this:

```
0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
```

To convert from binary to hex, first break the binary number up into groups of 4 bits. Each of these 4-bit groups will have a value from 0 to 15, so each group can be converted to a single hex digit, like this:

```
1111 1111 1111 1111
 F    F    F    F
```

Now put an "&H" in front of the hex number (&HFFFF) so the interpreter will know what we are talking about. To create a dashed line, the process would look like this:

```
1111 0000 1111 0000
 F    0    F    0
```

The PATTERN statement would then be used like so:

```
PATTERN &HF0F0
```

and all lines drawn will now be dashed. To return to the default solid lines, we would use the PATTERN statement again, like this:

```
PATTERN &HFFFF
```

and we're back where we started from, drawing solid lines.

Listings One and Two both use area fill patterns rather than line patterns to create dithering effects. Creating area fill patterns with the PATTERN statement is done a little differently. The syntax would be:

```
PATTERN ,PatternArray%
```

In this case we must provide the name of a short integer array as the argument, and there must be a comma after the PATTERN keyword, as shown.

The array, of course, must be defined first. It can be as large as necessary, as long as the number of elements it contains is a power of 2 (2, 4, 8, 16, etc.). Array elements begin their count at 0, so the actual DIM statements would use values such as 1, 3, 7, 15, etc.

The values assigned to the array elements are determined in the same way as the line patterns above. Each element corresponds to a single horizontal line of the pattern 16-pixels long. We could then imagine a simple checkerboard pattern like this:

```
1111000011110000
1111000011110000
0000111100001111
0000111100001111
```

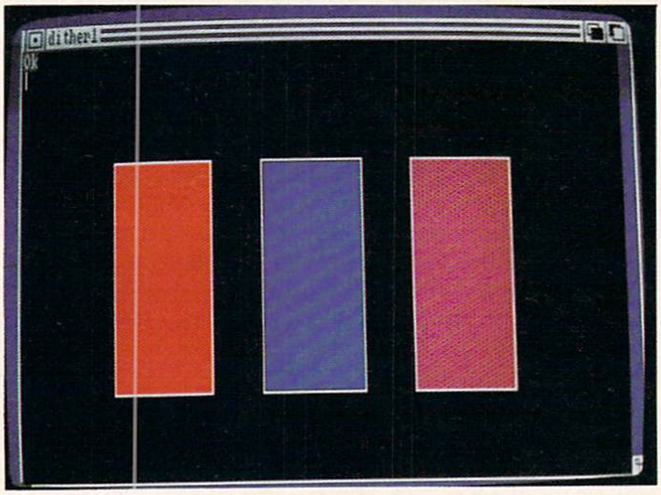
Bringing this to the screen would be done like so:

```
DIM P%(3)
P%(0)=&HF0F0
P%(1)=&H0F0F
P%(2)=&HF0F0
P%(3)=&H0F0F
PATTERN ,P%
```

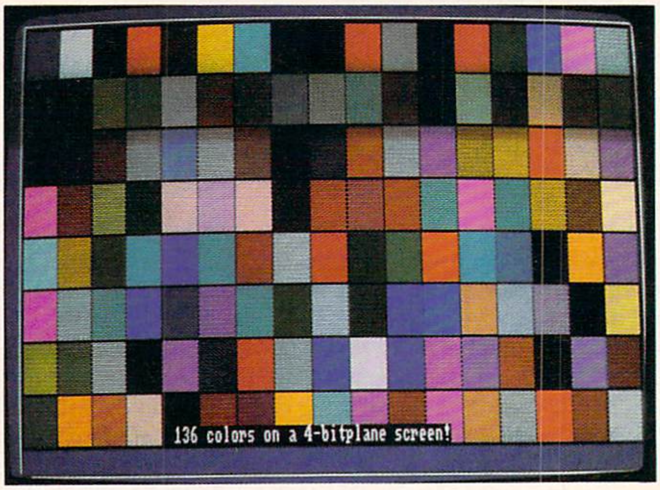
Each 16-pixel line of the pattern was converted from a binary number to a hex value and assigned to each element of the array. All area-filling operations, such as PAINT, would now be rendered in the above checkerboard pattern. To return to the solid fill pattern, dimension an array to 2 elements and assign both the value of "&HFFFF". Then call PATTERN again, using this new array as the parameter.

Now, if all we could do with dithering was mix existing palette colors with the background, it wouldn't be all that exciting. We would just end up with an additional shade for each color on the existing palette. Fortunately, there is more to be had from dithering.

Dithering really comes into its own when we can actually mix 2 or more of any of the palette colors to produce a completely new color. For example, if we could mix any 2 colors on a 2-bitplane (4-color) screen, we would have an apparent palette of 10 colors to choose from: 0, 1, 2, 3, 0+1, 0+2, 0+3, 1+2, 1+3, and 2+3. A screen with a depth of 3 would give us 36 colors, a depth of 4 would produce 136, and a 5-bitplane screen could provide us with a total of 528 colors. Carrying this even further, in Extra-Halfbrite mode (normally 64 colors) we could derive 2,080 choices, and a HAM Mode screen could theoretically produce 8,390,656 variations of color!



```
Ok
```



136 colors on a 4-bitplane screen!

Besides the increase in the number of displayable colors, dithering also offers a savings of memory resources which might be put to better use for other purposes. Bitplanes do consume RAM—from 8 to 32K or more each, depending on the resolution mode used. Also, this RAM must be CHIP RAM—the lower 512K, regardless of the amount of expansion memory provided. This same area of RAM must also be used for all image data otherwise required by your program, like graphic arrays. As good as AmigaBASIC is, it is not the most memory-efficient language, and graphics-intensive applications can eat up that CHIP RAM quickly. Dithering allows the programmer to add those few extra colors to a display without the added memory requirements of an extra bitplane.

How is dithering done? There are probably many ways, but Listing Two gives a relatively easy method. This demo listing produces 136 apparent colors on a 4-bitplane, hi-res screen which is theoretically capable of only 16 different colors. The rest are dithered combinations of colors 0 to 15. There are no LIBRARY calls involved, no arcane PEEKs and POKEs or fancy footwork—just straight AmigaBASIC.

Enter the listing and have a look at the results. This is not the fastest running program ever witnessed. Certainly it could be better optimized for quicker execution but, as its real value is as an instructional demonstration, I did not invest much time in greasing the rails. If you own the AC/BASIC Compiler from Absoft, it will compile directly as written and speed things up.

Some of the shades displayed are quite similar. All the apparent colors are formed by combining pairs of the true colors. In order to produce the maximum of shade variation when the colors were dithered, I tried to create a basic palette consisting of colors that were as dissimilar as possible. I do not doubt that someone with a better eye for color could come up with a more suitable starting palette for this purpose. In actual practice, though, I do not think color displays are enhanced by incorporating a lot of dissimilar colors. Subtle variations of existing hues provide a more pleasing and artistically "coherent" effect in my opinion, but feel free to experiment. Changing only one PALETTE statement will affect a number of the resulting apparent colors. Try it.

Once you've gotten the hang of using the PATTERN statement, dithering any two palette colors is very easy. All that's required is an alternating bit pattern, as shown in the first example, and the COLOR statement to control the current foreground and background colors. The foreground color will occupy the "1" bits, while the "0" bits will be of the current background color. Any two colors can then be combined to produce a new apparent color.

The actual dithering in Listing Two is done by two subprograms called ColorCycle and DitherFill. The former cycles through all the foreground/background color combinations possible with 16 colors. DitherFill does the actual area filling with the PAINT statement.

I chose a 640 x 200 screen with a depth of 4 for this demo. It seemed to be a good compromise in terms of the dithering effect produced and the complexity of the source code. Going to a 5-bitplane screen would require changing the resolution to a lo-res (320 x 200) display, thus increasing the pixel size. Larger pixels degrade the "illusion" of dithering, causing the colors to take on a polka-dot appearance. A lo-res interlace (320 x 400) screen would solve this but would also introduce flicker. Stepping up to Extra-Halfbrite or HAM modes requires more complex coding. I wanted to keep it as simple as possible.

Omnitek Computers
AMIGA
FULLY AUTHORIZED
SALES AND SERVICE
FOR ALL COMMODORE
PRODUCTS
SOFTWARE CLEARANCE SALE
20 - 75% OFF
We carry a complete line of software
and accessories for the AMIGA.
Come to our Salem, NH location and
pay NO SALES TAX!
1300 Main St.
Tewksbury, MA 01876
(508)851-4580
78 Main St.
Salem, NH 03079
(603)893-9791
Circle 128 on Reader Service card.

Even so, having a choice of 136 colors on a screen you thought could only squeeze out 16 is a definite improvement. You would probably be hard pressed to find a place for them all in most applications, except perhaps a paint program. I showed this demo to a friend who is a devout C programmer, without telling him how I did it. I told him only that it was a 4-bitplane screen. Talk about a double take! He was convinced I had somehow tricked him with an EHB or HAM mode display.

As stated earlier, the method outlined here is only one possible technique for creating dithering effects on the Amiga. Use of the PATTERN statement works well for simple area fills, but not all graphics are drawn this way. You might want to experiment with routines that render one pixel at a time with the PSET statement in a loop, which also switches the foreground color between each PSET execution. A sort of airbrush effect can be created this way.

Of course, dithering is not limited to the mixing of just two colors. Pixels of several different hues can be combined with very good effect if grouped closely together on a hi-res screen. And don't forget text applications. Unlike most other machines, the Amiga treats text displays as graphic objects, so they can be dithered too.

When combining different colors, how do you know what color you will end up with? The art of combining two or more colors to produce a different hue is a rather complex subject in itself. More information can be obtained from most any basic text on color photography. Some art texts also have information on this subject. But be aware that combining different colors of pigment produces an entirely different result than combining colors of light, as is done on a monitor. Make sure the reference covers the latter.

To get started, all you need to know is that red + green = yellow, red + blue = magenta, and blue + green = cyan. Also, any color can be darkened by adding dark gray or black, and lightened by adding light gray or white. See if you can come up with a dithering routine of your own. It's an area of Amiga graphics which has seen only a smattering (pun intended) of software application. Be the first kid on your block to come up with an 8-million-color HAM doodle, and the world will beat a path to your door.

(continued on page 74)

Pixels At An Exhibition: The Video

Amiga-generated art is showcased at a University of Vermont art show

by R. Shamms Mortier, PhD.

Pixels at an exhibition.
May I offer my rendition
Of a digitized suspicion
That we share a common mission
To unearth our superstitions
Via ocular submission?
Pixels at an exhibition.

The short poem above was printed on handouts that accompanied the exhibition of my Amiga videographics and electronic paintings at a November, 1988 show at the Coburn Gallery on the University of Vermont's Burlington campus. To my knowledge, the event represents the first time that microcomputer work—much less Amiga-generated work—was shown in a traditional gallery setting in that part of the Northeast.

I see the acceptance of this medium into what has often been an all too exclusive setting as a groundbreaking step. I'm confident that ten years from now shows containing computer graphics work and videographics will be available to the public all over the country. Computer paintings and videography will take their respective places alongside all other media, to be judged solely on the basis of their aesthetic merits.

For the moment, however, there still seems to be resistance to accepting computer-generated imagery as "serious art." It's taken me a long time to get even this far.

Eight years ago, as the manager of the graphics service at the University of Vermont, I could feel the revolutionary changes in graphics design coming. To prepare for these changes, I began to study the new technology by taking several computer programming courses. One of these, in which students were asked to design software for children with learning disabilities, gave me my first exposure to microcomputer graphics. I was hooked immediately.

The first systems that I worked on gave me no hint as to the pace at which changes in graphic design would really take place. In class we worked on Apples, whose jagged, eight-color electronic graphics—"drawn" on the attached graphics tablet—were magic to my eyes. Not long after that, I bought a Commodore 64, and began my honeymoon with Commodore graphics.

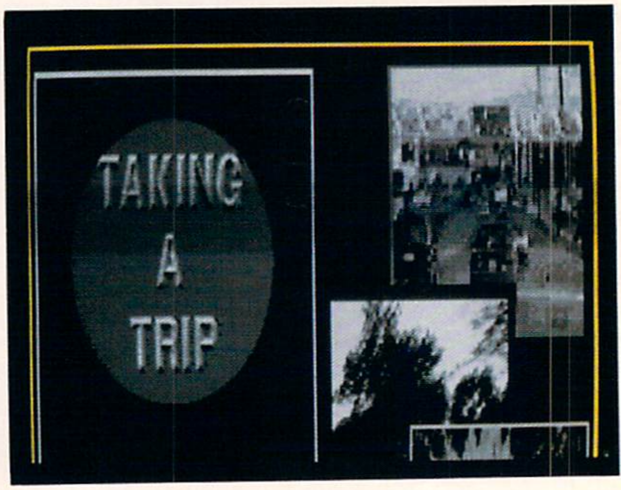
Meanwhile, attracted to the name and reputation of "Big Blue," the University consented to purchase an IBM for my use. Several thousand dollars later, I had a system that could produce sixteen-color charts and graphs (whoopee!). Unfortunately, it was of little use in bringing to fruition the more extensive instructional graphic designs and projects that I had in mind.

Then in 1985, after a serious eye operation forced me to argue my point from a darkened bedroom, I was finally able to persuade the skeptical media director—who was of the opinion that Commodore made only game machines—to authorize an investment in an Amiga 1000. (Unfortunately, Commodore's lack of advertising has never made it easy for those of us in education to convince those who hold the purse strings to buy Commodore products.) Not long after purchasing the Amiga for the office, I took funds from a personal business venture and bought an Amiga 1000 for my home studio.

Since that time, the Amiga has held its own, enjoying a number of placements on the campus. It has even invaded the sacrosanct halls of UVM's art department, where—with the help and support of a number of individuals—I was able to secure a Fall '88 showing for my personal Amiga work.

My show consisted of two parts: a videotape (which is the subject of this article) and a collection of seventy-five

Left
The Legend of Sleepy Hollow
Right
Taking an Amiga video trip



prints (which I will discuss more completely in a later article). I wanted the video to run continuously during the show, demonstrating to the audience the wide range of graphics possibilities on the Amiga.

Here is a description of the techniques I used to make the videotape, along with a list of the processes and software necessary for each...

Titles and Credits

I used Caligari (the professional pre-release version) to create interesting backgrounds for all of the titles and credits. Caligari has a fascinating object-design interface that gives you a state-of-the-art view of the 3D universe. I turned on my VCR and just played around with it on the screen, grabbing 3D structures with the mouse and saving everything for later editing.

The Caligari images are nice, but the Hi-Res flicker is a bit bothersome. However, considering that this was going to be genlocked into the background, I found the results satisfactory for my purposes. The spinning and rotating 3D primitives gave a certain high-tech look to the piece, and made an interesting background for titling text animations.

I first drew the titles and credits in DPaint II—mostly with the Kara Color Fonts. This technique gave them an almost classical look. By choosing the proper colors, I was able to make the flicker almost unnoticeable, even in Hi Res.

I then incorporated the drawings, one by one, into Video EFX 3D (from Innovision). This is one of my favorite titling EFX programs, because its results are very smooth and professional. With this software, it's easy for you to spin graphics up from infinity, or even give them 3D depth as they fly along. Video EFX 3D does have one drawback, however: If you use it, be prepared to wait as long as several hours for your final renderings.

Each of my interspersed titles and final credits reached its resting place on the screen differently. Some spun around the Y-axis like swiveling doors; some flew up from the infinite background, spinning and flopping wildly; and some drew so close to the screen that the pixels became tremendously exaggerated. In one particularly long editing session, I genlocked each of them against segments of the Caligari backdrop. The result was a collection of very interesting titling segments.

The Bit Bucket

COMPUTER STORE

We Want Your Business!!

We Have the Best Prices!!

Oldest Commodore Dealer in the Area!!

2 Locations to Serve You

* Software
* Hardware
* Service
* Information

1294 Washington Street
W. Newton MA 02165
617-964-3080

621 Boston Post Road
Sudbury MA 01776
508-443-9731

COMMODORE
AMIGA
Authorized Dealer

Authorized Commodore Amiga Dealer and Commodore Service Center

Circle 142 on Reader Service card.

The Paintings

After the opening titles, the next segment on the tape was a "slide show" of some of my electronic paintings. Some were duplicates of prints hanging on the wall; others were shown only on the videotape. Lasting about twelve minutes, this segment drew much attention. Swirls, wipes, slides, fades... One by one the static images transformed themselves from one form into the next.

What I hoped to do with this segment was to alert people to the fact that the works hanging in the gallery were indeed produced using a digital medium. It was also interesting to contrast the physical prints with their original videographic parents.

I also wanted to take advantage of the excellent slide-showing tools available in Lights/Camera/Action (by Aegis), the most versatile and user-friendly slide-showing program I have been able to find. The interface is carefully designed, with options that only an experienced media artist could conceive of. Yet, at the same time, it's friendly enough that sporadic users don't have to reference the manual constantly. The program also allows you to segue into ANIM files—though I didn't use it for that purpose this time around.

Sleepy Hollow

I was fortunate enough to have been asked by Richard Ramella, the editor and founder of JumpDisk magazine, to illustrate a series of Amiga paintings for Washington Irving's "The Legend of Sleepy Hollow." These ten paintings, along with Irving's text, were marketed by JumpDisk in October of 1988.

To fit the images, the text, and a viewing program onto my disk, I had to render them in Lo-Res non-overscan, making them quite different—in both concept and design—from the other works in my exhibition. Usually, my works are multi-thematic and rendered with PhotonPaint in HAM interlace. The contrast between my paintings and the relatively linear story of Ichabod Crane was so radical that I decided to incorporate Irving's masterpiece—and my thoughts on it—into the video.

Again, I used Lights/Camera/Action as the slide-show utility. Naturally, I made sure that the music for this segment was different from that used in

MASTERPIECE PROFESSIONAL FONT COLLECTION©

20 DISK SET

The largest collection of fonts and clip art available in a single package for the AMIGA.

110 DIFFERENT FONT STYLES

This doesn't mean 10 sizes of 11 fonts. It means 110 DIFFERENT fonts.

LARGE SIZES

Specially designed for video work. 95 % of the fonts are over 100 pt. tall. Easily resized smaller.

PATTERN CLIP ART

141 hi-res DPaint II pages. There are thousands of objects and examples.

ALL FONTS ARE HI-RES

BRUSHES - 2 disks full of color brushes.

COLORFONTS - 4 full disks.

100 PAGE MANUAL - Full size font printouts.

20 DISK SET - ONLY $199.00

Contact your local AMIGA dealer or order direct from
AROCK Computer Software, 1306 E. Sunshine,
Springfield, MO 65804 1-800-288-AROK

DPaint II is a registered trademark of Electronic Arts.

Circle 107 on Reader Service card.

the others. This segment was about two minutes long.

Taking a Trip

The title of this segment, "Taking a Trip," is somewhat of a double-entendre. (My peers from the sixties should readily grasp the more figurative of the two meanings.) If the title itself didn't make this fact clear to some at the show, then the visuals invariably did.

It took me several weeks to put this segment together. First, with a borrowed video camera sticking out the window of a '76 pickup, I filmed the thirty-five mile journey from Burlington, Vermont (where I work) to my home in Bristol. I held the camera as steady as I could, and just let it run. The result was quite interesting, and made me realize that I hadn't really looked at everything that I pass on my way to and from work each day.

Next, with the VCR running, I hooked up "video out" to a video utility amplifier (a CVA2B-4 from the Comprehensive Video Supply Corporation in Northvale, N.J.). This unit allowed me to split a video signal to two targets. I took one "out" line and connected it to the Amiga Live unit (from A-Squared), while the other I connected to the "video in" on my SuperGen genlock. By using the sliders on the genlock, I was able to fade in and out of various digital EFX produced by Live.

Unfortunately, Live has never really made good on their pledges to provide software support. They promised to make available—at a price of fifty dollars—four upgrades in one year's time. So far, they have delivered only a Beta test of the first upgrade (and it's between almost two years since they first made the offer!).

Luckily for Live, a company called Elan Design has twice saved their skins—first with InVision, and more recently with Elan Performer. InVision is one of the niftiest packages available for EFX work. It's designed to work with the Live digitizer, and it interfaces in ways that music-video producers really enjoy. If you have enough memory, the touch of a macro-key can give you any of the eighty EFX: digital mirroring, pixelating, colorizing, framing, and posterizing. In short, InVision really brings Live to life, making it well worth the investment.

With Live connected and InVision loaded up, I started to manipulate the video that I shot from the pickup truck. The results were something between state-of-the-art video EFX and—from what I'm told—a strange substance-abuse experience. For example, in the middle of a shot of passing trees and houses, the sky suddenly turned green and the whole scene mirrored itself on a horizontal axis. Then reality intervened again.

The main problem that I encountered in this segment was that Live reacted poorly to some of the InVision interventions. Once in a while the vertical hold would let go. However, it did work well as a background piece for the wandering audience. I further enhanced it using an Amiga musical composition from the soundtrack.

I probably could have been arrested for overusing EFX, but I wanted to do more than just hold a storyline together: I wanted to demonstrate to my audience the breadth of the possibilities that these tools offer. Admittedly, this segment may have been somewhat overrun, coming in at around thirteen minutes.

Mandalas

The mandala is a traditional visual device used in many cultures to help the mind concentrate during meditation. Using the Amiga, I was able to design and animate mandalas to my heart's content. The fruits of these labors made up the next segment of the video.

I chose DeluxePaint II as my painting tool, using it in a way that I like to think is original—although hundreds of other Amigans have probably used it in the same way. With the menu bars and the cursor hidden, I turned on color-cycling (with the TAB key) and the Symmetry Tool (from the Toolbox). Then, using various brushes, I alternately painted and erased areas on the screen. The results have to be seen to be believed. With the proper sitar sounds added to the soundtrack, a truly meditative experience was produced.

Some experimentation is warranted before this process is recorded, but I believe any Amiga artist can produce wonderful results here. For this reason alone, you should save your DP II—even if you upgrade to DP III. As far as I can tell, DP III doesn't allow you to hide the cursor, which you must do to produce animated mandalas.

The Family

The fifth segment of the video was called "The Family." It's basically a flip-book animation sequence using frames taken from a videotape of my family interacting at the breakfast table on a Sunday morning. To film the original video, I just turned on the camera and sat down in front of it with my wife and my children (who wandered in and out of the room periodically). After a little while, everyone forgot that the camera was on, and some nice vignettes ensued.

After viewing the tape, I edited out sections and digitized what was left using DigiView. In total, I kept about sixty frames, which I dumped into DigiPaint for further editing. I left the human figures black and white, but colorized the background and foreground, giving the frames an almost surreal look.

When my editing was complete, I used Pageflipper+FX (from MindWare International) to piece the segments together. I use PF+FX constantly as an animation editor and consider it one of the most valuable professional videographic tools in my Amiga library. With it, I was able to splice together my single frames freely, as well as store to and play back from RAM at very high speeds.

The end result of my two weeks of labor was a two-minute sequence that had the look of a futuristic penny-arcade movie. Although somewhat abstract, the images definitely told a story about breakfast at the Mortiers.

3D

No Amiga video demonstration is complete without something in 3D. As the time of the show drew near, I knew I had to produce at least one short 3D piece.

I decided to use Forms-In-Flight—which has some of the best shading routines I've ever seen—to render my images. With it, I sculpted and spun a smooth, spaceship-like form. Adding a light source produced finished images that were both "touchable" and believable. The results are most pleasing if rendered in Hi-Res, but it takes a while to get comfortable with the program itself. With FIF, I rendered and saved about twelve views of my sculpture.

Then, with pieces of this same form, I used DPaint II to fashion a floating space station on an asteroid. While turning on color-cycling, I noticed that the drawing cycled the smooth colors in a very unique fashion. Pleased with the effect, I decided to incorporate it into the segment.

I taped the background alone, toggling color-cycling on and off at intervals. I played this back as a genlocked background, running the spaceship around the foreground with the help of PageFlipper+FX. By fooling with the sliders on my SuperGen while all this was going on, I was able to create an interesting 3D piece that lasted about two minutes.

Th- Th- That's All Folks!

Well, that's all there was to it. When I finished, I had a forty-four-minute video to run at the gallery. The soundtrack on the tape was made up of a host of compositions, each created and recorded with Amiga sound-generating software. I recorded the tape on 1/2" VHS, and someday I'll tape it again onto 3/4" tape for better clarity. The monitor I used to show the video was not the best; next time, I'll take more care to get the best playback equipment.

In a welcome reversal of the circumstances of a few years back, new Amiga products are now being introduced constantly, and old ones are being upgraded. (If Dpaint III and some of the other recently released programs had been out at the time, I definitely would have used their capabilities.) Releases in the last six months have been very exciting for Amiga animators and videots like me, but if I wait until the latest software or hardware is out before starting a project, I'll wait forever. The time to begin is when the spirit talks.

In short, the show was an Amiga extravaganza. It was really well received throughout the two-week interval. The video always had onlookers, and to my surprise, I received as many favorable comments on it as I did on the prints. Many Amiga people stopped by, and those non-Amigans who were not converted were at least tempted.

For my part, I came away having learned an abundance of amazing new videographic techniques. (Nothing expedites the learning process like a deadline hanging over your head!) I watch the video every month or so, and I'm still amazed at what can be accomplished in a few sleepless nights with the Amiga. The next galleried showing of my work will benefit widely from this experience, and my next tape is already in production. •AC•

More than just a Hard Disk Backup Utility

ExpressCopy

Rapidly copies directories and files to floppy disk

FAST Copies directories and files from Hard Disk to Floppy Disk at up to 1 MegaByte per minute. Can format, verify, AND fill a new floppy disk with files in less time than it takes AmigaDOS to format a new disk!

SAFE Other Hard Disk Backup Programs this fast create NON standard disks that can only be used by their program. NOT ExpressCopy! ExpressCopy creates STD DOS disks that look just as if you had done a copy from your Hard Disk to floppy. If your Hard Disk failed, the backup disks can be used NORMALLY! This gives you a SAFE and EASY way to accesss important files you backed up.

Select files by their DateStamp, pattern matching, Archive Bit, and by source directory. All file attributes (DateStamp, Protection Bits, and FileComment) are retained. Options for setting the Archive bit for incremental backups, verifying the data written to floppy disk, and estimating the number of disks needed for the backup. Up to 4 copies of the backup disks can be created at a time, or disks can be pre-loaded in up to 4 disk drives. New disks are automatcially formatted and verified. Easy recovery if a bad diskette is found. Parameters to be used for backups can be saved in configuration files. Either Normal or Fast File compatible disks can be written. Specifically designed for effective multi-tasking. Backup restoration can be done using any file copy program, your favorite Directory Utility, or by ExpressCopy's Restore program.

ExpressCopy has no copy protection and can be used from both the CLI and Workbench.

Fully documented with a 65 page manual which includes a 25 page section with help and ideas on how you can better organize and manage your Hard Disk.

All these features and speed for ONLY:

$44.95 US

Express-Way Software, Inc.
PO Box 10290
Columbia, MO 65205-4005
(314) 474-2984

Requires an Amiga with at least 512k ram and Version 1.2 or higher of AmigaDOS
Amiga, AmigaDOS, and Workbench are trademarks of Commodore-Amiga, Inc.

Circle 122 on Reader Service card.

ARRAKIS

200 Armstrong Road
Las Cruces, NM
88001

BBS: 505-523-4405 24 Hours/Day
Voice: 505-526-6243 M-F 8AM-5PM MST

Abacus Software
ToolBox......................18.95
AssemPro....................55.95
Accolade
Jack Nickalaus Golf....28.50
Fast Break...................25.50
Antic Software
GFA Basic....................78.95
Phasar..........................50.50
Zoetrope Anim Sys......78.95
ASDG
Dual Ser Port............169.95
Bethesda
W Gretzky Hockey........28.50
BroderBund Software
SIM CITY.......................25.50
Byte by Byte
Sculpt 3-D....................56.50
Animate 3-D.................84.50
Centaur Software
Forms in Flight II..........67.50
BAD..............................28.50
World Atlas...................33.95
Central Coast Software
Quarterback.................39.95
DOS 2 DOS...................30.95
Disk to Disk..................28.50
Cinemaware
Lord of Rising Sun.......28.50
King of Chicago...........14.50
DesignLab
Fine Print......................28.50
Discovery Software
VIP................................28.50
ZOOM...........................16.95
Sword of Sodan...........28.50
Digitek
Hole in One Mini Golf...22.50
Epyx
500 XJ Joystick............12.95
Free Spirit Software
Ami-Align.....................28.50
Fuller Computer Systems
Project D........................8.50
Gold Disk
Comic Setter.................56.50
CS Data Disks (ALL)....19.95
PageSetter....................56.50

Professional Page......221.95
Professional Draw......112.50
Design 3-D...................56.50
Movie Setter.................56.50
Grandma's Software
NAG 3.0........................44.95
Haitex
Haicalc..........................34.95
X-Specs 3D..................69.95
Hypertek
GOMF 3.0 Button..........42.50
Incognito
Atredes 1.1 BBS Pro....84.50
Kingdom's of Eng........28.50
Opticks.......................112.50
Innovision Technology
Video Effects 3D.........112.50
Broadcast Titler..........168.50
Interactive Softworks
Calligrapher 2.0 ....72.95
Studio Fonts................26.50
Newsletter Fonts..........26.50
Lion's Fonts.................50.50
Asha's Fonts................50.50
Novelty Fonts (Callig)..39.50
Interplay
Battle Chess.................28.50
KFS Software
The Accountant..........168.50
Mastertronics
War in Middle Earth......28.50
Magic Johnson.............28.50
Michtron
HiSoft Basic.................89.95
HiSoft Dev Pack...........56.50
Micro Search
City Desk......................84.50
Perfect Sound..............50.50
Micro Systems Softwre
RAW Copy....................33.95
Micro Systems Software
Online Platinum............56.50
Scribble Platinum.........84.50
The Works....................84.50
The Works Plati..........165.95
Excellence..................168.50
Microillusions
Fire Power....................14.50

Land of Ledgends........28.50
Photon Paint 2.0 ....4.50
Photon Pt Surf Dsk.......6.95
Photon Vid Cell Anm....84.50
Music-X........................68.50
MindScape
Captain Blood...............28.50
Harrier Combat.............28.50
Indiana Jones Tmpl......22.50
MindWare
Page Flipper F/X...........89.95
Page Render 3-D..........89.95
Charon 5.......................22.50
New Horizons
ProWrite II.....................70.50
Flow.............................56.50
ProScript.......................28.50
ProFont (each).............19.95
New Tek
Digi-View Gold............112.50
Digi-Droid.....................44.95
Oxxi
ATalk 3.0.......................56.50
Maxiplan 500................84.50
MaxiPlan Plus.............112.50
Nimbus.........................84.50
Panasonic Inc
WV-1410 w/lens.........189.95
16 MM lens...................16.95
Par Software
Express Paint...............78.95
Easy Ledger...............166.95
Project Manager.........109.95
Precision
SuperBase Pers............44.95
SuperBase Pro 3.0 ....196.50
SuperPlan.....................84.50
Pro-Sound Designer.....89.95
Progressive Peripherals
IntroCAD.........................4.95
PixMate.........................39.95
Psygnosis
Barbarian......................22.50
Obliterator....................22.50
Terrapods.....................22.50
RainBird
The Guild of Thieves....25.50
Star Glider II..................25.50

Universal Mil Sim...........8.50
UMS Sim 1....................11.95
UMS Sim 2....................11.95
Carrier Command.........25.50
ReadySoft
A-Max.........................112.50
Dragon's Lair...............33.95
Ganymed.......................16.95
Right Answers Group
The Director.................39.50
Director ToolKit............22.50
Soft Logic
Page Stream...............112.50
PS Fonts (each)...........22.50
Software Heaven
Dungeon Master...........22.50
Software Visions
MicroFiche Filer...........56.50
Spectrum Holobyte
Falcon..........................28.50
Tetris............................19.95
Supra Corporation
Supra Modem 2400......99.95
Taito
Bubble Bobble..............22.50
Gladiator.......................22.50
Operation Wolf.............22.50
Qix...............................22.50
Rastan..........................22.50
Renegade.....................22.50
Sky Shark.....................22.50
Unison World
PrintMaster Plus...........22.50
Art Gallery (each).........16.95
Fonts & Borders...........19.95
William Hawes
ARexx..........................27.95
W-Shell.........................27.95
Zuma Group
TV Show.......................56.50
TV Text.........................56.50
Epyx JoySticks
500 XJ..........................12.95
GVP...............................Call!
Electronic Arts...............Call!
All brand names...... carried!
For the hottest specials call
the Arrakis BBS # !!!

Inside Atredes 1.1 VHS.............19.95

Supra 2400 Baud Modem........99.95

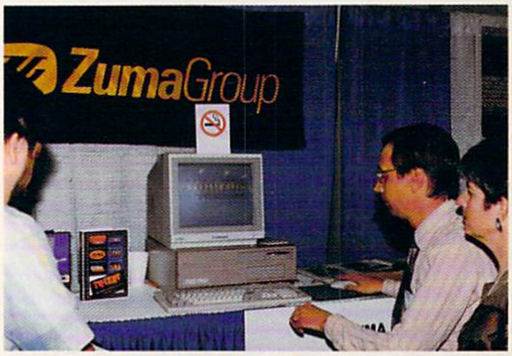
Epyx Joystick 500XJ..........12.95

ASDG Dual Serial Board........169.95

Circle 108 on Reader Service card

AmiEXPO

Chicago: July 28-30, 1989



Security Matters

The Amiga has often been called "The best kept secret in the computer industry." At AmiEXPO in Chicago, some users felt that this title was going to be strictly enforced. AmiEXPO attendees and exhibitors were scrutinized closely by powerful looking men and women in conservative suits. As the Amiga visitors moved through the lower halls of the Hyatt Regency, they passed dozens of uniformed officers and quite stern looking personnel with earphones and two way radios discretely hidden beneath their suits.

The reason for these special security precautions became clear as Amiga enthusiasts passed by the Grand Ballroom, a floor above the AmiEXPO. The National Governors Conference was being held at the Hyatt that weekend and the following week President Bush was scheduled to appear on Monday, a day after AmiEXPO closed. The reason for the security was suddenly obvious, yet it did not make it easy on Ms. Kim Morey, who was dressed in camouflaged combat fatigues and sporting a name tag that read "Mrs. Uzzi." She had to explain several times to the officers upstairs that she was demonstrating a new product at the Micro Momentum, Inc. booth downstairs, and that she posed no threat to national security.

Speakers and Classes

Seeing an Amiga celebrity or attending a special class on your favorite Amiga application are the highlights of any AmiEXPO. This show featured Jim Sachs, Computer Artist and Programmer; Dr. Henri Rubin, Chief Operating Officer of Commodore International Ltd.; and Tim Jenison, President of NewTek, Inc. Mr. Jenison entered the Saturday morning hall with roller skates on his feet and a gas-powered propeller on his back! (Mr. Jenison is known in his local community for commuting in this style at speeds of up to forty miles per hour!)

No exposition of Amiga enthusiasts would be complete without workshops, lectures, or displays. Once again, Joe Lowery and Steve Jacobs lined up an interesting assortment of Amiga notables for panel sessions and lectures. Some of the best three-dimensional art, animation, and special videos were effectively displayed, in an effort to excite more artists into the Amiga medium. AmiEXPO is the closest thing to a national Amiga users group, and our best opportunity to witness the achievements of our fellow Amiga users.

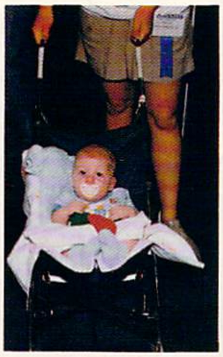
AmiEXPO attracts Amiga users of all persuasions!

Amiga Pioneers

Jay Miner assisted at the Intuitive Technologies booth.

Amiga Trendsetters

Tim Jenison NewTek, Inc.'s President shares his thoughts.



New Amigans

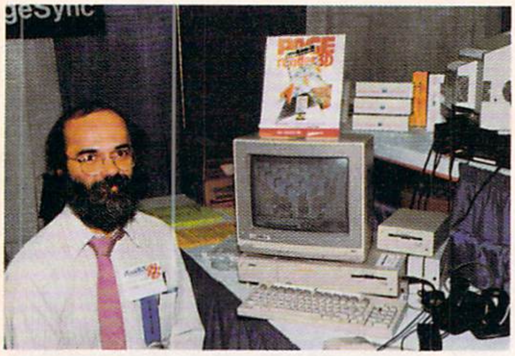
Alexx Morrison, at nine weeks old, appears to be the youngest Amigan.

New Amiga Companies

Todor Fay, of Blue Ribbon Bakery kept everyone well fed with homemade cookies.

Amiga Hardware?

Paul Montgomery With NewTek, Inc.'s newest hardware?



Show Floor Excitement

On the show floor there was enough excitement to keep all the Amiga enthusiasts happy. Hardware vendors sported faster and larger hard drives, new accelerator cards, and special networking equipment, while software developers rolled out their newest games, development languages, and productivity packages. Several dealers were offering better-than-average prices on software and hardware.

Great Valley Products Inc. once again occupied one of the largest booths, with a very classy, high-tech look. GVP was demonstrating numerous hard cards, including their removable media drive and two of their '030 board models (16 and 25 Mhz.). Company officials announced the new 68030 board with a 33 Mhz chip for delivery in 1 to 3 months. A 32 Mhz version was demonstrated which completed generating a complete hi-res ray-tracing in under 15 minutes. (The wireframe popped up in 3.14 seconds, while the scanline paint mode took a blazing 1 minute flat.) Also shown was their 150 meg streaming tape drive, which runs at 6 megs/minute and has three backup modes, including UNIX/tar.

The excited people from **Ready Soft, Inc.** were demonstrating their **AMax MacIntosh Emulator™** and showing demos of their new Amiga game, **Space Ace,** due for release by year's end. They also talked about Version 2 of the AMax which is scheduled to have hard drive support and possibly even AppleTalk support. (For additional AMax information, please refer to Interactive Video Systems.)

Interactive Video Systems demonstrated their **Trumpcard** products, as well as a shareware AMax patch that gives hard-drive support with their Trumpcard (utilities should be available for this in about a month.) The package looked extremely smooth during their demonstration on a 2000 with hard drive, flicker Fixer™ and a Zenith flat-screen monitor. This set up can run most Mac programs including Hypercard. They also showed a RAM card for their TC 500 which will take 256K or 1 meg Simms for a total of up to 4 meg.

NewTek, Inc. had a pre-programmed demo of their Toaster™ running during all three days of the show (still no hands-on demos.) With their Demo Reel 2 and the new Digi-Paint 3, **NewTek** had one of the largest and most impressive booths. The newest "unofficial" release for the Toaster™ is the first of the year.

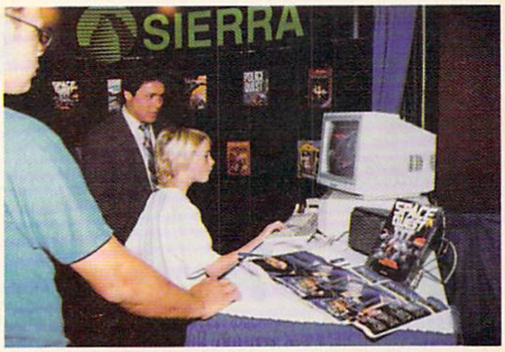
Micro Momentum, Inc., manufacturers of the Journeyman (the first portable Amiga), demoed their telecommunications tutorial software, **TeleTutor**, along with **Momentum Check**, a checkbook management program. For Amiga gamers there was the **Uzzi Interface**; a joystick adapter that accelerates the firing rate for some of the most exciting game action we've seen.

The nice folks at the **Zuma** booth were demonstrating their new **TV Text Pro™** which will probably be released by the time you read this. We were shown some remarkable titling effects including embossed wallpaper which can be locked into place so you can title over. Fonts can be extruded in 3D to adjustable depths. Other effects include glint, cycling, cast, drop, and transparent shadows. It can use any fonts, including Colorfonts.

Daniel Ten Ton of **Digital Animation Productions** in Boston, Mass. demonstrated his **Amiga-based video transputers**. According to Daniel, once their is a finished manufactured board, a special version of **Turbo Silver** will be written for his machine. The processing speed of the transputer can be increased by adding more processing boards. Rendering speeds of one frame per minute can be achieved with the proper ray-tracing software.

Elan Design had a small booth, but they drew a big crowd with demos of their **Invision** video effects system and **Performer**.

Sierra on Line was showing **Space Quest III**, **Leisure Suit Larry II**, **Silpheed**, **Police Quest II**, and the newest **King's Quest** all of which will be released for the Amiga in November.

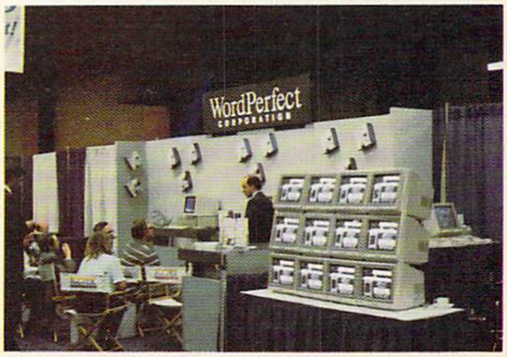


Perky Melissa Grey, President of **Blue Ribbon Bakery**, and Todor Fay put on their baker outfits and roamed the crowds with large baskets of home-baked cookies, blowing the diets of all. At their blue-ribboned Amigas, they demonstrated their **Who! What! When! Where!** organization software. Melissa, a musician herself, demonstrated and discussed their forthcoming **Bars and Pipes** (due in October), which she describes as the first object-oriented music system. She says that it deals with music, not MIDI-information. According to Melissa, "you provide the creativity...we provide the musical knowledge!"

CMI showed their processor/accelerator board, as well as their new multiport networking board, which is **AppleTalk™ compatible**. The engineers on duty in the booth said that a few file-protocol software handlers were needed to make the network fully compatible with Macintoshes, but the system was working fine with direct connections to Amigas.

Mindscape, demonstrating their various games, announced the August release of their newest **Fiendish Freddy's Big Top o' Fun**.

Sylvester Lee of **Lee Software** seemed to draw a lot of attention with his new **Total Control Diet** software, which not only helps Amigans keep fit and trim, but keeps track of the calories and nutrition of what you eat, with the help of a database of over 2400 foods.



Supra Corporation was showing their RAM boards, modems and hard drives.

VidTech International demonstrated their new **Scanlock** genlock, announced as a broadcast-quality output device at a consumer price.

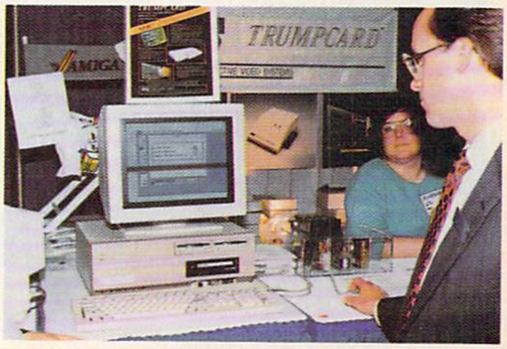
Central Coast Software, makers of Quarterback, showed their latest: **Mac 2 Dos**, a file transfer utility which lets you read and write Mac disks on your Amiga.

A real standout of the Expo was the **Intuitive Technologies** booth. Mike Lehman, creator of **ULTRACARD**, was showing off his product all weekend with the help of Jay Miner, who seemed truly fascinated by the program. Lucky attendees were able to have their ULTRACARD purchases autographed by Jay.

Two of the largest booths at the show belonged to retailers. Dr. Oxide was there with his **Comp-U-Save** team, while Frank Khulusi was there with his crew from **Creative Computers** of California. Both booths were packed for most of the show.

Bob Maudzinsky from **Mindware**, makers of **PageFlipper Plus F/X** and **Page Render 3D** showed these products and discussed their upcoming **Page Sync** a music synchronization program.

Psygnosis Ltd. made a few whispered promises about future Amiga products to be made available in October and November. Their booth was crowded with the array of titles they currently publish.



Thank You Chicago!
Amazing Computing collected $574.34 in contributions (in exchange for the current copy of AC) to be given to the American Cancer Society of Chicago.

For more information on vendors' products or announcments, please use the inquiry numbers on pages 60 and 61, and circle them on the Reader Service Card by page 96.

AC talks with the "Father of the Amiga"

A Conversation with Jay Miner

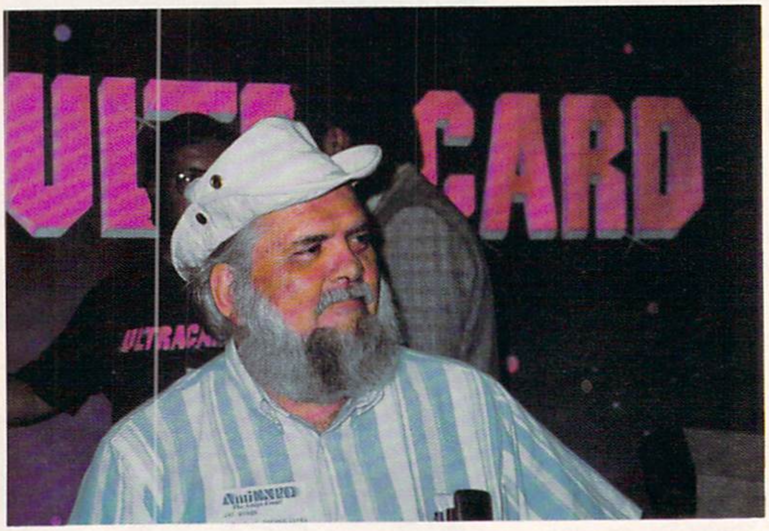
by Steve Gillmor

The 1989 AmiEXPO in Chicago was a study in contrasts. The number of vendors seemed down a bit from the previous show in New York, yet the attendance was strong. There were a few new products exhibited, including several hardware and software developments that filled crucial holes in the Amiga's superstructure.

The advance of ARexx in the Amiga environment was signalled by the release of Digi-Paint 3, the first presentation of Mindware's TASS ARexx-driven application operating system, and the launching of UltraCard. Much will be written and discussed about UltraCard and hypermedia in the coming months, but AmiEXPO Chicago '89 will go down as the event at which the promise of an easy-to-use, multitasking, ARexx-friendly Amiga was realized.

Nowhere was this driven home more clearly than at the UltraCard booth, where the architect and team leader of the Amiga's pioneering hardware design, Jay Miner, was stationed. Waves of fans clustered around "Padre" (as he is known on Plink and other networks), getting his autograph and words of wisdom on the state of the Amiga. Afterwards, Jay and I sat down and tried to put things in perspective:

The "Father of the Amiga" was on hand at AmiEXPO Chicago to meet with Amiga zealots



AC: How did this AmiEXPO feel to you?

Jay: It seemed a little smaller than last year—physically that is. I didn't really get time to wander around. I was too busy in the UltraCard booth.

AC: I am certainly interested in your UltraCard thoughts. But before that, I'd like to know how you feel about the current state of things on the Amiga?

Jay: I still think it's the best computer around in the home PC class. However, I am still bothered by the lack of advertising of it as anything else than an artist's machine.

AC: Do you feel it can compete effectively at the work-station level?

Jay: Certainly, it is architecturally a superior machine. We still don't have the software base of the others, but that is rapidly catching up.

AC: Are the others catching up to us in hardware?

Jay: In specs and appearances they are, but from what I hear the graphics and animations and sound are all bottlenecks that are slow and cumbersome.

AC: What kinds of programs would you like to see advertised more?

Jay: They need to show programs like PageStream, City Desk, Superbase and WordPerfect as being used, and better on the Amiga, because of the superior windows and multitasking.

AC: What is your basic configuration?

Jay: I have a 2000, with a GVP 80 MB hard card, and an 8 MB memory card, with 4 MB loaded. Also, I have a flickerFixer, which I can't overrecommend, and a beautiful Zenith flat-faced monitor that has the most lovely colors and contrast!

AC: Where do you see holes in Amiga software?

Jay: Oh, good question. [Pause] There was a big hole in the HyperCard area but that is now being filled by UltraCard. I hear that the desktop publishing is slightly inferior to others, but it is still very useful.

AC: Okay, let's talk about the HyperCard area. Why do you feel that's such an important type of program?

Jay: Because it allows "non-programmers" to construct Graphics "shells" that can access and control other programs without having to learn a language like C or M2.

Going back to the subject of flickerFixer for a moment, I would like to recommend that Commodore install one with the Zenith flat-faced monitor in every dealer's store in the United States—especially at shows.

AC: With UltraCard as a front end?

Jay: Really! UltraCard should be given out with each Amiga like the Mac does. It makes a tremendously good-looking display, and will sell more Amigas than anything else I can think of.

AC: How did you get involved with UltraCard?

Jay: I was at a FAUG [First Amiga Users Group] meeting a month ago, and Mike [Mike Lehman, author of UltraCard] demonstrated it there. After the demo, all anyone could ask was, "Yes, that's nice, but what does it do?"

I could see he needed some help, so I offered to help him improve his Demo and tutorial. Then he said he needed help at the AmiEXPO booth, so I worked with his program for a week, to become familiar with it, and went to Chicago to help in the booth.

AC: What feedback or comments did you get from people at the show about UltraCard?

Jay: Most people said it was about time the Amiga had a version of HyperCard. Others still complained about the demo, which we didn't have time to improve much. It was hard to explain that the graphics were only an example of what could be done, and not part of the program itself.

AC: Jay, you have been quoted in the past as being quite concerned about the continued survival of the Amiga. As the "Father of the Amiga", how do you feel today?

Jay: I feel Commodore should try harder to support the developers, and not to compete with them so much. Also, I am very concerned with the present trend to advertise the Amiga solely as an artist's machine, and neglect the great programs in desktop publishing, word processing, spreadsheets, and database management. They are building up the impression of the Amiga as a graphics computer only, and that hurts the long range marketing of the Amiga very much.

I like the hardware work they have been doing, the 68020 and '030 cards, and the new machines. I just hope they are working on a complete revision of the chip set, done in CMOS, with full 32-bit chip address range. It can be done now with the new chip design rules, and would result in a chip set that would make a lovely portable downstream.

AC: Can you give us some examples of what UltraCard stacks can be, and need to be, constructed?

Jay: There is one example that comes to mind. Imagine an exhibition hall filled with Amigas (well, 10 to 20), that each show a map of all the exhibits, with icons for their locations (among the potted palms.) Now imagine clicking on an exhibit icon and having a picture of the exhibit pop up, with speech describing the exhibit, and then clicking on the people in the picture, and having the summary of the person shown, and clicking on boxes, that bring up details of the exhibit.

AC: What do you tell people are the advantages of UltraCard?

Jay: The main advantages of UltraCard are also the advantages of the Amiga, such as real multitasking, ARexx interfacing and, of course, also the color graphics, speech, and sound. ARexx is an Amiga Rexx language that allows programs like UltraCard to talk to each other and control each other. If you have the ARexx libraries in your Libs directory then UltraCard can act as a Graphics shell to control and exchange data with any other program running simultaneously. This is why ARexx and the Amiga's multitasking make UltraCard such a powerful tool, and it's much easier to program a graphics controller using UltraTalk than C or M2.

AC: More powerful than HyperCard?

Jay: Yes, much more powerful. HyperCard has neither multitasking, nor ARexx, not to mention color and multiple resolutions.

AC: So instead of a subset of several programs' features, you can have the access to a superapplication of the best of all worlds—

Jay: Yes, that's right. One application I would like is to use it to design a graphics controller for a dialer that pulls data out of MicroFiche Filer (another ARexx-interfaced program) and does autodialing for me.

AC: It's available! [See "Adventures In ARexx" in *Amazing Computing* V4.6.] I will send you the ARexx macros—

Jay: Great, thanks.

AC: After I tack UltraCard on top of it, unless you do it for me, Jay.

Jay: So you can do a custom dialer controlled with UltraCard. Also UltraCard has built-in UltraTalk, like HyperTalk.

AC: Yes, Mike Lehman kept the two languages similar enough that UltraCard users could take advantage of the many books and stacks available using HyperCard.

Jay: Yes, that similarity will allow someone to do a future translator.

AC: On a personal note, how is your health?

Jay: I am generally in good health, except for this darned cold I caught here in Chicago. Completely lost my voice. My new kidney that my sister gave me is working fine, I have been swimming and water-skiing on my boat in the delta for months now, and feeling great.

AC: Do you anticipate being more active in the Amiga community again?

Jay: Oh definitely. I hope to find work one of these days in that community, and I am presently very active in networking

Zorro II
Prototyping Board

- * Over 4400 Plated Holes on a 0.1" Grid.
- * Gold Plated Edge Connector.
- * "D"-type I/O Connector Pattern.
- * Accepts 64 Pin DIPs and 14x14 PGAs.
- * Low Inductance Power and Ground Pattern for High-Speed Designs.
- * Designed for Maximum Flexibility.
- * Includes Mounting Bracket.

To order, send:
check or money order for $49.95 +
local sales tax (California only) +
shipping & handling (US: $3.00, Foreign: $6.50)
in US dollars to:

Celestial Systems
Suite #165
Box J
Manhattan Beach, CA 90266
(213) 372-1229

Circle 112 on Reader Service card.

in the Amiga community. People ask me, "What kind of next computer would you build?" and I say, "If I had such a great design in my head (which I might have) why should I tell the world?"

AC: But you'll tell me though—

Jay: Maybe not the truth!

AC: So help you God?

Jay: Isn't that the same as telling the world?

AC: Of course. Is there anything else you would like to pass along to the *Amazing* Amiga Community, Jay?

Jay: Oh yes, I would like them to "advertise" the Amiga. Do this by buying an extra copy of an Amiga magazine, like *Amazing*, and leave one—only one per Amiga fan is needed to saturate—in every doctor's office and waiting room in the country. If Commodore won't do it, we have to. Or use your old copies of such magazines if you can bear to part with them. OK?

AC: OK.

AC: Jay, thanks for talking with us, and keep in touch, OK?

Jay: I sure will—on Plink, Compuserve and PORTAL.

•AC•

AmiEXPO

ABACUS SOFTWARE
5370 52nd Street
Grand Rapids, MI 49508
(616) 698-0330
Inquiry # 206

A.C.E.
P.O. Box 591
Oak Forest, IL 60452
(312) 687-7788
Inquiry # 207

ADVANCED GRAVIS COMPUTER TECHNOLOGY LTD.
6894 Palm Avenue
Burnaby, BC V5J 4M3 Canada
(604) 434-7274
Inquiry # 208

AMERICAN PEOPLE/LINK
165 N. Canal, Suite 650
Chicago, IL 60606
(312) 648-0660
Inquiry # 209

AMUSE, NEW YORK AMIGA USERS
151 1st Avenue, Suite 182
New York, NY 10003
(212) 460-8067
Inquiry # 210

BLUE RIBBON BAKERY
1248 Clairmont Road, Suite 3D
Atlanta, GA 30030
(404) 377-1514
Inquiry # 211

C LTD.
723 East Skinner
Wichita, KS 67211
(316) 267-3807
Inquiry # 212

CALIFORNIA FREEWARE
1747 East Avenue Q, Suite A-3
Palmdale, CA 93550
(805) 273-0300
Inquiry # 213

CENTAUR SOFTWARE
14040 Tahiti Way, Suite 528
Marina Del Ray, CA 90292
(213) 821-5995
Inquiry # 214

CENTRAL COAST SOFTWARE
424 Vista Avenue
Golden, CO 80401
(303) 526-1030
Inquiry # 215

CHICAGO COMPUTER SOCIETY
P.O. Box 8681
Chicago, IL 60680-8681
(312) 794-7737
Inquiry # 216

COMP-U-SAVE
414 Maple Avenue
Westbury, NY 11590
(516) 977-6707
Inquiry # 217

CREATIVE COMPUTERS
4453 Redondo Beach
Lawndale, CA 90260
(213) 370-2009
Inquiry # 218

DIGITAL ANIMATION PRODUCTIONS
79 Mount Vernon Street
Boston, MA 02108
(617) 720-2038
Inquiry # 219

DIGITAL WORLD
711 Army Trail Road
Addison, IL 60101
(312) 543-9000
Inquiry # 220

ELAN DESIGN, INC.
11 Upper Terrace, CH
San Francisco, CA 94117
(415) 621-8673
Inquiry # 221

FOX VALLEY AMIGA USERS GROUP
1408 West Briarcliff
Bolingbrook, IL 60439
(312) 887-2303
Inquiry # 222

FULLER COMPUTER SYSTEMS
P.O. Box 9222
Mesa, AZ 85204-0430
(602) 835-5018
Inquiry # 223

GATEWAY AMIGA USERS GROUP
P.O. Box 811
Bridgeton, MO 63044
(314) 739-5181
Inquiry # 224

GRAFX COMPUTING
1140 Post Road
Fairfield, CT 06430
(203) 255-5166
Inquiry # 225

Chicago '89

Vendors, Distributors, and Users Groups

GREAT VALLEY PRODUCTS
225 Plank Road
Paoli, PA 19411
(215) 889-9411
Inquiry # 226

HAITEX RESOURCES
208 Carrollton Park, Suite 1207
Carrollton, TX 75006
(803) 881-7518
Inquiry # 227

INNERPRISE SOFTWARE, INC.
128 Cockeysville
Hunt Valley, MD 21030
(301) 785-2266
Inquiry # 228

INTERACTIVE VIDEO SYSTEMS
15201 Santa Gertrudes #Y102
La Miranda, CA 90638
(714) 944-4443
Inquiry # 229

LAKE FOREST LOGIC, INC.
28101 E. Ballard Road
Lake Forest, IL 60045
(312) 816-6666
Inquiry # 230

LATTICE, INC.
2500 S. Highland Avenue
Lombard, IL 60148
(312) 916-1600
Inquiry # 231

LEE SOFTWARE
2881 Stoneridge Drive
Garland, TX 75044
(214) 530-5337
Inquiry # 232

MEMORY AND STORAGE TECHNOLOGY
3881 Benatar Way, Ste. E
Chico, CA 95928
(916) 342-6278
Inquiry # 233

MICRO ED ENTERPRISES, INC.
444 N. Orleans Street Suite 250
Chicago, IL 60610
(312) 245-0066
Inquiry # 234

MICRO MOMENTUM, INC.
100 Brown Avenue
Johnston, RI 02919
(401) 949-5310
Inquiry # 235

MICRO SYSTEMS INTERNATIONAL
1143 Monroe Street
Carleton, MI 48117
(313) 654-8402
Inquiry # 236

MICRO-SYSTEMS SOFTWARE
12798 Forest Hill Blvd, Suite 202
West Palm Beach, FL 33414
(407) 790-0770
Inquiry # 237

MICROSEARCH
9896 Southwest Freeway
Houston, TX 77074
(713) 988-2818
Inquiry # 238

MICROTECH SOLUTIONS, INC.
991A Aurora Avenue
Aurora, IL 60505
(312) 892-3551
Inquiry # 239

MINDSCAPE INC.
3444 Dundee Road
Northbrook, IL 60062
(312) 480-7667
Inquiry # 240

MINDWARE INTERNATIONAL
110 Dunlop W Box 22158
Barrie, ON L4M 5R3 Canada
(705) 737-5998
Inquiry # 241

NEWTEK
115 W. Crane Street
Topeka, KS 66606
(913) 354-1146
Inquiry # 242

PSYGNOSIS, LTD.
P.O. Box 483
Addison, Il 60101
(312) 620-4444
Inquiry # 243

READYSOFT INC.
30 Wertham Court, Unit 2
Richmond Hill, Ontario L4B 1B9 Canada
(416) 731-4175
Inquiry # 244

SHARP ELECTRONICS CORPORATION
Sharp Plaza
Mahwah, NJ 07430
(201) 529-9500
Inquiry # 245

SIERRA ON-LINE
P.O. Box 485
Coarsegold, CA 93614
(209) 683-6858
Inquiry # 246

SOFT-LOGIK PUBLISHING CORP.
11131 F South Towne Square
St. Louis, MO 63123
(314) 894-8608
Inquiry # 247

SPIRIT TECHNOLOGY CORP.
220 W. 2950 South
Salt Lake City, UT 84115
(801) 485-4233
Inquiry # 248

SUPRA CORPORATION
1133 Commercial Way
Albany, OR 97321
(503) 967-9075
Inquiry # 249

TRANSACTOR/AMIGA USER
85 West Wilmot ST, Unit 10
Richmond Hill, ON L4B 1K7 Canada
(416) 764-5274
Inquiry # 250

VIDTECH INTERNATIONAL
2822 NW 79th Avenue
Miami, FL 33122
(305) 477-2228
Inquiry # 251

VISIONARY LTD.
45 Whithorn Crescent
North York, ON M2J 3B1 Canada
(416) 497-0833
Inquiry # 252

WORDPERFECT CORPORATION
1555 North Technology Way
Orem, UT 84057
(801) 225-5000
Inquiry # 253

XETEC, INC.
2804 Arnold Road
Salina, KS 67401
(913) 827- 0685
Inquiry # 254

ZUMA GROUP
6733 North Black Canyon Highway
Phoenix, AZ 85015
(602) 246-4238
Inquiry # 255

Walk All Over Your Amiga

Simple Cell Animation In Modula-2

by Nicholas Cirasella

When people refer to graphics and animation on the Amiga, they invariably refer to blitter operations. The blitter is the workhorse for many graphics library routines like Bobs and VSprites. Instead of accessing the blitter directly, many programs utilize the Gel graphics routines, which in turn make use of the blitter.

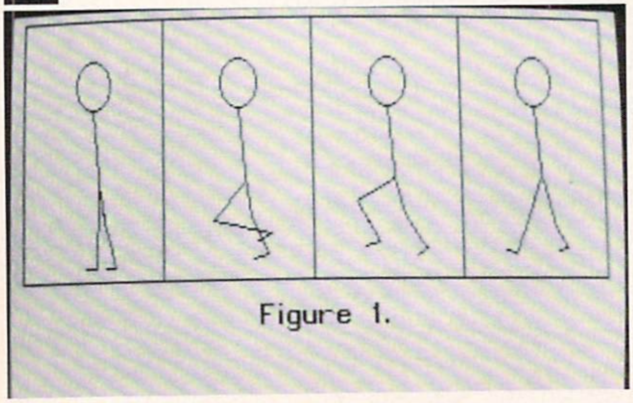
I find that many beginning and intermediate programmers would like to incorporate moving images or animation into their programs, but they are put off by the many complicated aspects of Gel routines. The thought of initializing Gel structures, setting Bob pointers and flags, and sorting and drawing Gel lists is enough to discourage all but the most ardent of developers.

As an alternative to the more complicated Bob animation routines, there are some very simple blitter routines in the ROM graphics library which allow direct access to the blitter's ability to transfer data. Taking advantage of this fact, I would like to present a programming example in Benchmark Modula-2 which will show how the blitter can be used to produce simple cell animations. This sample program illustrates techniques you can use to make the animated figure of a man walk across your screen.

About the blitter

The blitter is a special graphics processor designed to move data around in memory. In this example, you will be moving data (pixel images) from a bitmap to your RastPort (drawing area). The blitter routine BltBitMapRastPort (which will be discussed later) will do this job for you nicely. Other useful blitter routines include BltBitMap, which is used to copy image data between bitmaps, and ClipBlit, which is used to copy image data between RastPorts. These routines can all be imported from the Blit Module found in the M2L directory supplied with your Benchmark Modula-2 system.

Figure One:
Body positions that make up a simulated walking motion.



Cell animation

For those of you who do not know what cell animation is, a quick explanation is in order. In short, it is a method of animation in which multiple images are quickly overlapped on top of one another. Since each cell image is slightly different from the one before it (e.g., an arm may be raised slightly higher), an illusion of motion is produced. The most common examples of this type of animation are cartoons and children's books in which a picture is drawn at the top right corner of every page. Flipping through the pages generates a short animation sequence. This method of animation is perfectly suited for the Amiga, as the blitter is designed to move images quickly.

To run the sample program, you will first need to create some cell images with a paint program. Start by turning on the Coordinate Display feature of your paint program. Then, on a 320 x 200 resolution screen, draw a hollow rectangle in the upper left-hand corner with dimensions of 300 x 91 pixels. Next, draw vertical lines through the rectangle, spacing them exactly 75 pixels apart. This should leave you with four adjacent boxes (cells) of equal area. Before continuing, check the coordinates of each one. The coordinates of the top left corner of Cell 1 (not to be confused with the top left corner of the border you drew) should be (1, 1), while those of the bottom right corner should be (74, 90).

To create your animated figure, draw it first in Cell 1. Next, cut this image, copy it into Cell 2, and alter it appropriately. Repeat these steps, copying the newly altered image into Cell 3 and altering it yet again. After performing the same operations between Cells 3 and 4, you should be left with a different image in each of the four cells. Be sure to leave at least 5 pixels of blank background to the right of the image in each cell; this spacing will be necessary when you later animate your figure. Figure 1 illustrates the body positions that make up a simulated walking motion.

You may either save your set of cells as a picture file or, to conserve disk space, cut the set and save it as a brush. The sample program requires the file to be named Cells.pic and expects to find it in the current directory at execution time.

The source code

The sample program CellAnimations.mod consists of only three procedures. The first procedure, Initialize, is used to load your cell images into a bitmap (CellBM, in our example) for storage. It also opens your screen and window. The RastPort and ViewPort pointers are set to point to the proper places while the LoadRGB4 routine is called to load the palette of our cell-image file into the current ViewPort.

The second procedure, DrawMan, is the routine that handles the actual cell animation. It renders each cell in sequence, proceeding from Cell 1 to Cell 4, and then back to Cell 1 to repeat the cycle. The important variables in this procedure are CellLength, which is set to a value equal to the width of your cells (in pixels) plus one (remember, if you change the size of your cells you must change this value accordingly); CellPos, which holds the cell number currently being rendered; SourceX, into which the calculated X-coordinate of each cell rectangle is placed; and Xoffset, which is the X-coordinate of the current cell to be drawn.

The Drawman procedure begins with a loop that moves your figure across the screen twice.

```
SetRast(rp^,0);
```

renders background pen 0 into every pixel of your screen, effectively clearing the viewing area.

```
FOR i := 1 TO 36 DO
```

is the loop that controls how many times your cell images are drawn.

```
INC(CellPos,1);
```

loops through the number of the current cell to be drawn.

```
INC(Xoffset,-5);
```

subtracts 5 from the X-coordinate of the last cell drawn. To experiment with varying the increments of your animation, try subtracting values higher and lower than 5.

```
IF CellPos=5 THEN CellPos:=1; END;
```

cycles back to Cell number 1 when the last cell is reached.

```
SourceX:= (CellPos-1) * CellLength + 1;
```

calculates the X-coordinate of the start of each cell. CellPos contains the number of the cell to be drawn.

WaitBOVP(vp^) is called just prior to the calling of the blitter routine. This routine is used to synchronize the reading of your image with the screen-updating feature of the Amiga. Calling this routine will help prevent the image of your animated figure from flickering.

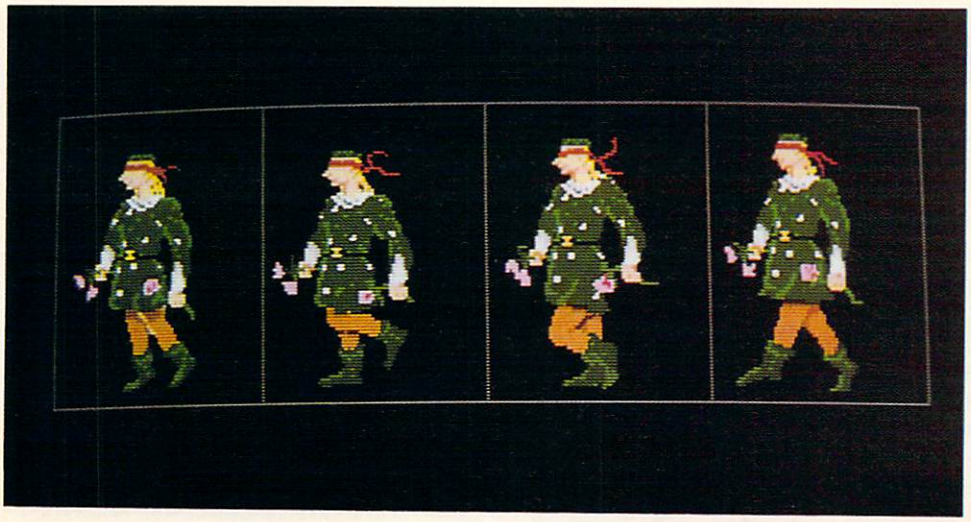
Finally, there is the BltBitMapRastPort routine. It takes the following form:

```
BltBitMapRastPort(VAR SourceBM : BitMap; SrcX, SrcY :
INTEGER; VAR DestRP : RastPort; Destx, DestY : INTEGER;
SizeX, SizeY : INTEGER; minterm : MinTermFlagsSet);
```

SourceBM is our source bitmap which we will load into our image containing our animation cells; SrcX, SrcY is the X-Y coordinate offset into the source bitmap; DestRP points to your destination Rastport; DestX, DestY is the X-Y coordinate offset into your RastPort; and minterm is the value defining the type of image drawing you want (a value of 192 decimal will produce a vanilla copy).

Now, take a close look at each parameter as it is used in the program. Descriptions are contained within brackets:

BltBitMapRastPort(CellBM [the bitmap you loaded your cell into], SourceX [the calculated X-coordinate of cells from your source bitmap], 1 [the Y-coordinate of cells from your source bitmap], rp^ [the pointer to your RastPort], Xoffset [the counter of the X-coordinate to place the current cell onto your RastPort], 50 [the Y-coordinate to place the current cell onto your RastPort], 74 [the length of each cell], 90 [the width of each cell], and MinTermFlagsSet(192); [function that does an exact copy of source onto destination when given a value of 192].



Elfen magic. Four frames of cell animation drawn in your favorite paint program.

Corrections

A number of errors and inaccuracies appeared in the article *Exploring Amiga Disk Structures* (V4.6). The article's author, David Martin, provided us with the neccesary corrections and clarifications. The following are the corrections were brought to our attention:

A math error: 4,294,967,295 is the maximum value that can be stored in a 32-bit word (p. 59).

AmigaDOS disks use 1760 blocks, not 1758. Blocks zero and one are reserved by AmigaDOS for the AmigaDOS boot blocks. The first disk block available for user data is block two (p. 59).

Unused slots in the bitmap table are set to one, not to zero (p. 59).

The bitmap information is not stored in the root block, but in separate disk blocks. The bitmap portion of the root block contains pointers to these blocks (p. 60).

File notes are stored in a file header block and each subdirectory's user block. They cannot be squeezed into the user directory blocks (p. 60).

Each block's checksum is used to tell AmigaDOS whether the data is good when it is read later. It can be calculated easily: sum all the words in the block, ignore overflow, and subtract the result from zero. The resulting value—which should be zero if the data is good—replaces the initial zero in word five (p. 59).

The Archive bit, which tells the user that a file has been changed, tells backup programs whether the current version of a file has been backed up. The Script bit allows an AmigaDOS batch file to be executed without the user having to issue the EXECUTE command from CLI (p. 60).

Parent directory pointers are used by AmigaDOS for stepping back one level toward the root directory. Word 124 in the file header block points to the next entry in the hash chain. Word 125 in the same block points to the parent directory of the file (p. 60).

In the *Inside ULTRACARD* article (V4.7) and in subsequent references to it in later issues, we referred to ULTRACARD as the first HyperText for the Amiga. We were incorrect in that THINKER, distributed by Poor Person Software, was actually the first Hypertext for the Amiga. Although both programs use Hypertext, the main difference between them is that THINKER is text-oriented, while ULTRACARD is graphics-oriented.

We apologize for these errors and any difficulties or inconveniences they may have caused.

When this command is used, it creates a vanilla copy the same size and shape as your cell rectangles. The background color (pen 0) of each cell is drawn over any image on the screen, totally "erasing" it.

In this example, each cell is drawn 5 pixels to the left of the previous cell. This shift gives the illusion of motion, whereas the cycling of cells gives the illusion of animation. When you created your cells, you left at least 5 pixels of background data on the right side of each cell, so that the only part of each cell not overlapped by the overlaid cell would be imageless background data.

The last step in the DrawMan procedure is a loop that slows down the cell renderings long enough for you to view each image properly.

ShutDown, the third—and final—procedure, simply frees the memory allocated for your bitmap, which contained your cell images. The window and screen are then closed and the program terminates.

Possible enhancements

This sample program is a simple one, and it could easily be modified to include routines for background restorations. In many instances, a background picture—which should not be altered by your cell renderings—will be drawn into your screen's bitmap. In order to retain background image integrity, copy the rectangle of background data—where your first cell is to be placed—into a temporary bitmap. Your first cell is drawn into the rectangular screen area you just copied. To animate your figure properly, the background data is then rendered back into the screen's bitmap and the process is repeated for all the cells of the animation. (If you animate an object that does not require your moving it across the screen—such as a rotating planet or a windmill—then you need not be concerned with background restoration.)

This small sample program provides just a glimpse of how you can use blitter commands to perform various types of graphics animation. I suggest experimenting with this source code to learn more. Try changing the values of the minterm in the BltBitMapRastPort routine to see what they produce. Before you know it, you'll be walking all over your Amiga!

Listing One

```
(**********************************************************)
(*                                                        *)
(*     Cell Animations            Nicholas Cirasella      *)
(*                                                        *)
(*           Compiled with Benchmark Modula-2             *)
(*                                                        *)
(*     This is a sample program to illustrate blitter     *)
(*     commands to animate a figure across our screen     *)
(**********************************************************)

MODULE CellAnimation;

FROM AmigaDOS       IMPORT ModeOldFile,FileHandle,Open,Close;
FROM Blit IMPORT    BltBitMapRastPort,MinTermFlagsSet;
FROM Graphics       IMPORT BitMap,BitMapPtr;
FROM IFF            IMPORT IFFDone;
FROM Intuition      IMPORT ScreenPtr, CloseScreen, WindowPtr,
                     CloseWindow,WindowFlagsSet, WindowFlags,
                     ScreenFlags, ScreenFlagsSet;
FROM Rasters         IMPORT RastPortPtr,SetRast;
FROM ReadPict        IMPORT ILBMFrame, ReadPicture;
(* ReadPict is a module from the IFF Libraries *)
FROM RemAlloc        IMPORT ChipAlloc, RemFree;
FROM SimpleScreens IMPORT CreateScreen, ScreenBitMap;
FROM SimpleWindows IMPORT CreateWindow, WindowBitMap;
(* SimpleScreens/SimpleWindows are from the Simplified Amiga
   Library Module *)

FROM SYSTEM          IMPORT ADR;
FROM Terminal        IMPORT WriteString, WriteLn;
FROM Views           IMPORT ViewPortPtr, LoadRGB4, WaitBOVP;

CONST
  (* main window flag definitions *)
  WinFlags = WindowFlagsSet{Activate,Borderless};

VAR  (* these are Global variables *)
  winPtr : WindowPtr;     (* pointer to window structure    *)
  CellBM : BitMap;        (* bitmap to contain cell images  *)
  scr    : ScreenPtr;     (* pointer to screen structure    *)
  rp     : RastPortPtr;   (* pointer to Rast Port structure *)
  vp     : ViewPortPtr;   (* pointer to View Port structure *)

(*********************************************************)
PROCEDURE Initialize;
(*********************************************************)
VAR
  FHandle : FileHandle;   (* contains info to open our file*)
  frame   : ILBMFrame;    (* contains info about our file's
                             ColorMap *)

BEGIN
   (** Load cells of walking man **)
    FHandle := Open(ADR("cells.pic"), ModeOldFile);
    IF FHandle # NIL THEN
       IF ReadPicture(FHandle, CellBM, frame,
                      ChipAlloc) = IFFDone THEN END;
       Close(FHandle);
    ELSE
       WriteString("Can not load IFF file"); WriteLn;
       HALT;
    END;
    scr := CreateScreen(320,200,5,NIL); (* low res screen*)
    winPtr := CreateWindow(0,0,320,200,NIL,WinFlags,
              NIL,scr,NIL);

    rp := winPtr^.RPort;    (* set up RastPort pointer*)
    vp := ADR(scr^.ViewPort); (* set up ViewPort pointer*)

    LoadRGB4(vp^, ADR(frame.colorMap),SIZE(frame.colorMap));
    (* this sets our viewport to contain the same pen
       colors as used in the cell IFF file *)
END Initialize;

(*********************************************************)
PROCEDURE DrawMan;
(*********************************************************)
VAR
  SourceX,i,loop,CellPos,Xoffset,j : INTEGER;

CONST
  CellLength = 75; (* width of each cell + 1 *)

BEGIN
 FOR j := 1 TO 2 DO      (* move our man across screen twice*)
   SetRast(rp^,0);       (* clear our RastPort *)
   CellPos:=0; Xoffset:=220;
   FOR i:= 1 TO 36 DO
      INC(CellPos,1);    (* CellPos contains the # of the
                            current cell to draw *)
      INC(Xoffset,-5); (* move man left 5 pixels each loop *)
      IF CellPos=5 THEN CellPos:=1; END;
      SourceX:= (CellPos-1) * CellLength + 1;
```

Remember...

Dedicated to the *AMIGA*

Nothing but the best!

COMMODORE AMIGA

The Memory Location

396 Washington Street
Wellesley, MA 02181
(617) 237 6846

Call the *AMIGA* Experts!

Authorized Amiga graphic design dealer.
Our crew has over 30 years total combined experience with the Amiga!
We ship UPS Mon. thru Fri.
Store hours: 10-6 Mon.-Thr.
10-8 Fri. 9-5 Sat.
Commodore authorized repair in store.
Low flat rate plus parts. Warranty service.

Circle 143 on Reader Service card.

```
      (* this calculates X coordinate for each cell *)

      WaitBOVP(vp^);     (* this reduces flicker of image *)

      BltBitMapRastPort(CellBM,SourceX,1,rp^,Xoffset,50,74,
                        90,MinTermFlagsSet(192));

      FOR loop := 1 TO 20000 DO END;
   END;
  END;
END DrawMan;

(*********************************************************)
PROCEDURE ShutDown;
(*********************************************************)
BEGIN
   RemFree(CellBM.Planes[0]);
   CloseWindow(winPtr^);
   CloseScreen(scr^);
END ShutDown;

(*********************************************************)
BEGIN  (****************  MAIN  ************************)
(*********************************************************)

Initialize; (* call routine to set up screen, window, load
               our cells, and set up RastPort and ViewPort *)

DrawMan;    (* call to routine to draw our cells *)

ShutDown;   (* de-allocate memory and close window/screen *)

END CellAnimation.
```

•AC•

AMZ 9-89

Order Toll Free
800-558-0003

Since 1982
ComputAbility
Consumer Electronics

Order Toll Free
800-558-0003

IVS Trumpcard Hard Drive Packages for A2000 Series

Seagate

ST-157N 49 MEG	519 DEL
ST-177N 60 MEG	749 DEL
ST-277N-1 60 MEG	619 DEL
ST-296N 80 MEG	649 DEL

Quantum

40 MEG SCSI Pro Drive	629 DEL
80 MEG SCSI Pro Drive	949 DEL
100 MEG SCSI Pro Drive	1029 DEL

these kits include IVS Trumpcard SCSI hard drive ctrl, cable, software and delivery in the contiguous USA This is not anassembly kit ! It is a package.

Hard Drive Cards (A-2000)

Seagate

ST-157N 49 MEG	539 DEL
ST-177N 60 MEG	769 DEL

Quantum

40 MEG SCSI Pro Drive	649 DEL
80 MEG SCSI Pro Drive	969 DEL
100 MEG SCSI Pro Drive	1049 DEL

these cards include IVS Trumpcard ctrlr, mounting brkt, cbl, software, and FREEdelivery in the contiguous USA.

Modems

Amiga 1680 Modem	99
Avatex 1200E	69
Avatex 2400	139
Avatex 2400E	109
Avatex 9600 External NEW!	839
Supra 2400zi Internal (A2000)	139

Trumpcard 500 Hard Drive Packages for Amiga 500

Seagate

ST-157N-49 Meg	579 Delivered!
ST-177N-60 Meg	799 Delivered!

Quantum

40 Meg SCSI Pro Drive	679 DELIVERED
80 Meg SCSI Pro Drive	999 DELIVERED
100 Meg SCSI Pro Drive	1079 DELIVERED

These Packages include Trumpcard 500 enclosure, IVS Trumpcard ,SCSI hd drive Ctrl, Software & free delivery in the contiguous USA

Trumpcard 500 Ram Expander available-
512k • 1 MEG • 2 MEG • 4 MEG

Seagate SUPER SALE

ST-125N 20 Meg - SCSI	279
ST-157N 49 Meg - SCSI	349
ST-177N 60 Meg - SCSI	569
ST-225N 20 Meg - SCSI	265
ST-251N 40 Meg - SCSI	375
ST-277N 60 Meg - SCSI	415
ST-296N 80 Meg - SCSI	479

Quantum

40 Meg SCSI Pro Drive	529
80 Meg SCSI Pro Drive	899
100 Meg SCSI Pro Drive	929

XETEC Fast Card

Hard Drive Packages for Amiga 2000

Seagate

ST-157N-49 Meg	519 Del
ST-177N-60 Meg	749 Del

Quantum

40 Meg SCSI Pro Drive	629 DEL
80 Meg SCSI Pro Drive	949 DEL
100 Meg SCSI Pro Drive	1029 DEL

Card features Xetec's DMAx hard, autoboot ROM,autoconfig circuitry.. 25 pin SCSI conn., disk, utilities manual.

XETEC *Fast Card System*

Hard Drive Packages for Amiga 500

Seagate

ST-157N-49 Meg	699 Del
ST-177N-60 Meg	919 Del

Quantum

40 Meg SCSI Pro Drive	819 DEL
80 Meg SCSI Pro Drive	1139 DEL
100 Meg SCSI Pro Drive	1219 DEL

System includesAdaptor with autoboot ROM, enclosure, manual, Fast Trak disk with utilities & shielded SCSI cable. With power supply and fan

AUTHORIZED
COMMODORE
AMIGA™
DEALER

MASTER 3A Disk Drive

1010 Compatible $135

*Free Delivery to the Contiguous States

VIDEO PACKAGE

PANASONIC 1410 CAMERA
16MM LENS WITH VARIABLE IRIS
COPYSTAND WITH LIGHTS
DIGIVIEW GOLD
$419 DELIVERED!

GOLD DISK SCANNER

$899 DELIVERED
Free Delivery in the Contiguous 48 States

PRINTERS

Panasonic 1180 NEW NEW NEW	189
Panasonic 1191 NEW NEW NEW	229
Panasonic 1124 NEW 24 Pin	**329**
Star NX-1000	165
Star NX-1000 Rainbow (color)	229
Star NX-2400	299

Supra 2400 Ext. Modem & Cable

Free Delivery to the 48 Contiguous States

• $129 •

IVS INFINIT 40

Removable Media Cartridge Drive Packages

INFINIT40/I	1199
INFINIT40S *	1299
INFINIT40D *	1399

* Requires Trumpcard or Trumpcard 500

AMIGA SOFTWARE

1943	25.95
3D Options	31.95
3D Pool	22.95
A-Talk 3	64.95
Advantage, The	46.95
After Burner	31.95
Airball	24.95
Alien Syndrome	24.95
Alternate Reality : City	26.95
AmigaDOS ToolBox	38.95
Android Decision	23.95
Anim. 4-D	379.95
Anim 4-D Jr.	87.95
Anim. Editor	39.95
Animation Multiplane	52.95
Animation Rotoscope	48.95
Animator Apprentice	174.95
Annals of Rome	23.95
Anti Virus	24.95
Architectural Design Disk	22.95
Arexx	29.95
Arkanoids	18.95
Aunt Artic Adven.	25.95
Artura	25.95
Attack on London	22.95
Award Maker	32.95
Awesome Arcade Pack	32.95
Axe of Rage	31.95
B.A.D.	29.95
Baal	19.95
Balance of Power 1990	31.95
Bard's Tale	31.95
Hint Disk	15.95
Bards Tale II	38.95
Batman	28.95
Battle Chess	31.95
Battle Hawks 1942	31.95
Battletech:The Cr Hk Inc.	31.95
Baud Bandit	30.95
Berzerk 3	22.95
Bionic Commando	24.95
Black Jack Academy	24.95
Blasteroids	CALL
Blood Money	25.95
Boomeraid	25.95
Bride of the Robot	24.95
Broadcast Titler	184.95
Bubble Bobble	22.95
Bubble Ghost	22.95
Calamus	274.95
California Games	31.95
Caligari- Consumer	144.95
Calligrapher	74.95
Capone	24.95
Captain Blood	31.95
Captain Fizz	19.95
Carmen SanDiego/ Time	28.95
Carrier Command	28.95
Centerfold Squares	18.95
Chessmaster 2000	28.95
Chessmaster 2100	31.95
Chronoquest	32.95
City Desk 2.0	114.95
Clue Master Detective	25.95
Colony	31.95
Combat Course	25.95
Comic Setter	59.95
Comic Art Disks (each)	20.95
Contra	25.95
Crystal Clear Turbo	17.95
Curse Buster	19.95
Cyber Complex	21.95
Dark Side	24.95
Data Retrieve Pro	179.95
Data Storm	24.95
Deathbringer	24.95
Defender Of The Crown	31.95
Deja Vu: 1 or 2	31.95
Deluxe Music Constr 2.0	61.95
Deluxe Paint III	97.95
Deluxe PhotoLab	97.95
Deluxe Print II	51.95
Deluxe Productions	128.95
Deluxe Video V1.2	84.95
Demon's Winter	26.95
Denaris	24.95
Design 3D	59.95
Designasaurus	30.95
Devon Aire Diamond Caper	31.95
Digidroid	59.95
Digipaint3.0	64.95
Digiview Gold	124.95
Digiwork 3D	82.95
Dinosaur Discovery Kit	25.95
Disk 2 Disk	28.95
Disk Master	28.95
Disk Mechanic	54.95
Dos 2 Dos	31.95
Double Dragon	25.95
Doug's Math Aquarium	46.95
Dragon Force	32.95
Dragon's Lair	33.95
Dungeon Master	24.95
Dungeon Quest	31.95
Dynamic Drums	46.95
Earl Weaver Baseball	31.95
Commissioners Disk	14.95
MLBPA Stats	14.95
Elan Performer	35.95
Eliminator	24.95
Empire	31.95
Excellence 1 Meg	158.95
Exolon	22.95
Express Paint 3.0	79.95
F-40 Pursuit Sim.	28.95
FA/18 Interceptor	31.95
Facc II	20.95
Faery Tale Adven	30.95
Falcon	31.95
Fantavision	38.95
Fast Break	28.95
Fat Tracks	CALL
Federation	31.95
Ferrari Formula 1	31.95
Fiendish Freddy'	31.95
Final Assult	14.95
Fine Print	32.95
Fire and Forget	24.95
Fire Zone	22.95
Flight Simulator II	31.95
European Scenery Dsk	19.95
Flow	59.95
Form In Flight	69.95
Formation	44.95
Front Page	74.95
Future Design Disk	22.95
Future Sound 500	89.95
Galactic Conqueror	28.95
Games : Winter Ed	31.95
Gauntlet II	31.95
Gettysburg	38.95
GFA Basic 3.0	87.95
Gold of the Realm	24.95
Gold Runner 2	24.95
Gold Rush	24.95
Grabbit	18.95
Grand Prix Circuit	31.95
Graphics Studio	38.95
Gunship	35.95
Hardball	28.95
Harley Davidson	31.95
Harpoon	25.95
Harrier Combat Simulator	31.95
Heavy Metal	28.95
Heros of the Lance	26.95
Hole In One	21.95
Hole in One - Course Disk	13.95
Hostage	29.95
Human Design Disk	22.95
Hunt For Red October	31.95
Hybris	24.95
Hyper Dome	22.95
I Ludicrous	22.95
Impossible Mission 2	14.95
Indiana Jones	24.95
Indoor Sports	30.95
Innovatools 2	48.95
Instant Music	13.95
Inter Font	69.95
Interchange	28.95
IntroCad	46.95
Investors Advantage	59.95
Invision Plus	CALL
It Came From The Desert	31.95
Jack Nicklaus Golf	31.95
JN Courses of 1989	14.95
Jet	31.95
Jinks	18.95
Joe Blade	21.95
Journey	31.95
Jug	24.95
Kampfgruppe	38.95
Karafonts Headlines 1 or 2	44.95
Karafonts Subhead	44.95
Kind Words V2.0	64.95
King Arthur's Quest/Excal.	31.95
King's Quest IV	31.95
Kingdoms of England	31.95
Knights of Legend	31.95
Kristal, The	31.95
Lancelot	26.95
Last Inca, The	24.95
Leaderboard duo pack	22.95
Leatherneck	24.95
LED Storm	24.95
Leisure Suit Larry	25.95
Leisure Suit Larry II	31.95
Life and Death	31.95
Logicworks	184.95
Lords of the Rising Sun	31.95
Macro Paint	54.95
Magellan 1.1	114.95
Magic Johnson 512	24.95
Magic Johnson 1 Meg	31.95
Manhunter - New York	31.95
Marble Madness	13.95
Mavis Beacon Typing	31.95
MediaLine Clip Art	21.95
Menace	19.95
Michton Hit Disk #1	32.95
Micro Fiche Filer Plus	114.95
Microbot Design Disk	22.95
Midi Magic	CALL
Midi Rec Studio V1.1	39.95
Midi View	23.95
Mighty Nerds	31.95
Mindroll	19.95
Mixed Up Mother Goose	19.95
Moebius	38.95
Movie Setter	59.95
Music Studio 2.0	49.95
Music X	184.95
My Paint Data Disk	17.95
Netherworld	22.95
Night Force	28.95
Novelty Fonts	48.95
Obliterator	25.95
Off Shore Warrior	24.95
On Line Platinum	58.95
Operation Clean Streets	25.95
Operation Wolf	22.95
Optics	114.95
Oswald	23.95
Outrun	31.95
Page Flipper Fx/Page Sync	CALL
Page Renderer 3-D	94.95
Page Setter	59.95
Page Stream	127.95
Page Thinker	59.95
Pen Pal	84.95
Phaser	57.95
Photon Paint 2.0	94.95
Photon Paint Surface	19.95
Photon Video : Cell Anim.	87.95
Photon Video Transport	180.95
Pioneer Plague	25.95
Planet of Lust	24.95
Platoon	28.95
Pocket Rockets	25.95
Police Quest 1 OR 2	31.95
Populous	32.95
Power Windows 2.5	54.95
Powerdrome	26.95
Precisely	48.95
Print Master Plus	24.95
Pro Script	28.95
Pro Soccer	22.95
Pro Sound Design	99.95
Pro Video Gold	177.95
Pro Write 2.0	69.95
Pro-Draw	112.95
Profesional Football Sim	21.95
Professional Page	228.95
Structured Clip Art	36.95
Templates	36.95
Profonts 1 & 2	21.95
Project D+ Backup	30.95
Project Master	129.95
Prospector Mazes of Zor	25.95
Pub Games	25.95
Publisher's Choice	114.95
Publisher Plus	64.95
Puzzle Storybook	25.95
Quick Silver	41.95
Qix	22.95
Quarter Back	42.95
Questron II	32.95
Rambo III	22.95
Rampage	24.95
Rastan	25.95
Raw Copy	36.95
Realm of the Warlock	21.95
Reel Fish'n	29.95
Renegade	22.95
Return To Atlantis	13.95
Rhyming Notebook	26.95
Ring Side	24.95
Ringling Bro's Circus	24.95
Road Challenge	24.95
Road Raider	24.95
Roadwar 2000	26.95
Robocop	28.95
Rocket Ranger	31.95
Roger Rabbit	28.95
Romantic Enct @ Dome	25.95
Rubicon Alliance	20.95
RVF Honda	24.95
Savage	24.95
Scavenger	CALL
Scrabble	26.96
Scribble Platinum	87.95
Sex Vixens-Outer Space	25.95
Shinobi	31.95
Shogun	31.95
Side Arms	24.95
Silent Service	24.95
Sim City	28.95
Skate Wars	31.95
Sky Blaster	18.95
Sky Chase	24.95
Sky Fox II	13.95
Sky Shark	22.95
Slipped Stream	17.95
Softwood File II	62.95
Sorceror Lord	23.95
Space Ace	34.95
Space Harrier	31.95
Space Quest 1 or 2	31.95
Space Quest 3	CALL
Space Racer	19.95
Speedball	24.95
Spinworld	18.95
Star Glider II	28.95
Star Ray	29.95
Star Trek The Final Front.	36.95
Star Wars	24.95
Street Fighter	25.95
Strip Poker II	25.95
Sub Battle Sim	14.95
Super Back	48.95
Superbase Personal	46.95
Superbase Personal 2	87.95
SuperBase Pro 3.0	189.95
Superplan	87.95
Superstar Ice Hockey	31.95
Sword of Sodan	31.95
T.V. Sports Basketball	31.95
T.V. Sports Football	31.95
T.V. Text Professional	87.95
Talespin	31.95
Technocop	31.95
Temple of Doom	31.95
Tetraquest	24.95
Tetris	20.95
Text Ed Plus	48.95
Test Drive 2 - The Duel	28.95
California Chall.	14.95
Supercars	14.95
Thexder	22.95
Thinker	57.95
Thunder Blade	31.95
Time and Magik	26.95
Times of Lore	24.95
Titan	28.95
Tower Toppler	31.95
Tracers	22.95
Transcripts	41.95
Trials of Honor	32.95
Turbo	19.95
Turbo Silver 3.0 1 Meg	114.95
TV Show	64.95
TV Text	64.95
Twilight Zone	24.95
Ultima 4	38.95
Ultima III	24.95
Undersea Comando	31.95
Universe 3	31.95
Universal Military Simulator	31.95
Vampire's Empire	28.95
Video Effects 3-D	113.95
Video Fonts	28.95
Video Page Titler	86.95
VideoScape 3D 2.0	115.95
Interior Design	22.95
Vindicators	CALL
VIP Professional	59.95
Virus Infection Protection	28.95
Vista 3-D	87.95
W Shell	29.95
War In Middle Earth	31.95
Wayne Gretzky Hockey	31.95
Weird Dreams	24.95
Who, What, When	64.95
Willow	25.95
Wind Walker	25.95
Wings Of Fury	25.95
WordPerfect	134.95
Works Platinum, The	CALL
World Atlas	35.95
World Class Leader Board	28.95
World Tour Golf	13.95
X-Specs	84.95
Zak McKracken	29.95
Zany Golf	26.95
Zoetrope	87.95
Zork Zero	38.95
Zynaps	22.95

MISC. HARDWARE

Ami GEN	135
Super GEN	669
Magni 4004 Genlock	1389
Flicker Fixer	455
Perfect Sound.	69
Chinon FB 354 Int Drive	99
Color Splitter	89
Gravis Joysticks	31.95
IMG Scan	114

MasterCard
VISA

Since 1982
ComputAbility
Consumer Electronics
NO CREDIT CARD SURCHARGE

Order Toll Free
800-558-0003
WI ORDERS AND INFORMATION
414-357-8181 FAX 414-357-7814
P.O. BOX 17882 Milwaukee, WI 53217

ORDERING INFORMATION: **Specify system.** For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School P.O.'s welcome. **C.O.D. charges are $4.00.** In Continental U.S.A. include $3.00 for software orders 5% shipping for hardware, minimum $5.00. MasterCard and Visa orders please include card #, expiration date and signature. WI residents please include 5% sales tax. HI, AK, FPO, APO,Puerto Rico and Canadian orders, please add 6% shipping, **minimum $6.00.** All other **foreign orders add 15% shipping, minimum $15.00.** All orders shipped outside the Continental U.S.A. are shipped first class insured U.S. mail. If foreign shipping charges exceed the minimum amount, you will be charged the additional amount All goods are new and include factory warranty. We do not guarantee compatability & version #s. Due to our low prices all sales are final. **All Defective returns must have a return authorization number.** Please call (414) 357-8181 to obtain an R.A. # or your return will not be accepted. Prices and availability subject to change without notice. Shipping and handling are non-refundable. We ship the latest versions available to us. updates must be handled by end user directly with the manufacturer.

Circle 115 on Reader Service card.

Roomers

by The Bandito

[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third party source from whispers inside the industry. At press time, they remain uncomfirmed and are printed for entertainment value only. Accordingly, the staff and associates of* Amazing Computing™ *cannot be held responsible for the reports made in this column.]

The Amiga LIVE! story

It's been a good year for inside stories—first Epyx's secret project, and now the previously untold tale of how Amiga LIVE! came into existence.

Our story begins long ago, before the Amiga was released. A-Squared Systems came up with a design for a real-time video digitizer for the Amiga. They envisioned applications like video performance, publishing, and so on.

Commodore thought this was a great idea, especially since it promised to be cheaper and better than any IBM or Macintosh digitizers. So Commodore made A-Squared a deal whereby Commodore would market the product through its own distribution channels as a Commodore-labeled device.

Everything seemed fine. Prototypes were shown at trade shows, and much interest was generated. Then Digi-View came out, and when Commodore saw one of the HAM images created with Digi-View they said, "We want LIVE! to do this!" So Commodore took over the design of LIVE! and proceeded to work on it in their own labs. *[Editor's note: A-Squared claims that the initial idea to incorporate HAM software into LIVE! was its own, and that Commodore—which had already been given the LIVE! design to work on—refused to do so, fearing it would delay its release.]* But the designer they had working on LIVE! left, and more time passed until they found a new engineer.

By this time, Digi-View had been on the market for quite a while. Things were in turmoil at Commodore, what with changing chief executives and shifting marketing plans. Poor LIVE! kept getting shunted aside, being shown at trade shows, but never with any idea of when it would ship.

Finally, A-Squared managed to get Commodore to give them back the LIVE! design (much altered by this time) so A-Squared could bring the product to market. After some further development (which consisted mainly of undoing Commodore's engineering attempts), Amiga LIVE! for the Amiga 1000 finally went on sale—just in time for Commodore to stop producing the 1000 and announce the Amiga 500 and 2000, neither of which was compatible with Amiga LIVE!

So A-Squared had to redesign the product yet again, and after many months Amiga LIVE! was finally in marketable form. *[Editor's note: According to A-Squared, Commodore was obligated by contract to produce the Amiga 500 and 2000 versions of LIVE! When it did not undertake this development, A-Squared took it upon itself to do so.]*

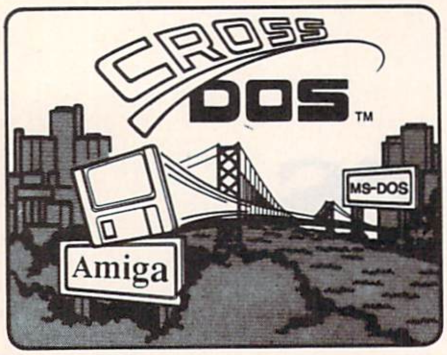
The latest story development is that A-Squared has won more than $890,000 in a lawsuit against Commodore. A-Squared charged that, by failing to market Live!, Commodore failed to live up to the terms of their agreement, and an arbitration board (confirmed by a judge) ruled in A-Squared's favor. So A-Squared will finally get some recompense for the years of battling Commodore, though it's not really enough to cover the lost market opportunity. *[Editor's note: According to Commodore, no payments have yet been made to A-Squared, and the ruling is currently under appeal.]* In many ways, the whole story is a warning of the pitfalls a small company can encounter when it tries to work with a giant manufacturer? Sometimes it's better to just stay in the garage and putter away.

1989 Developers' Conference Report

Of course, the Bandito's agents were in full force at DevCon '89. By carefully bugging the water glasses and planting microphones in the ceilings, they were able to gather much valuable data.

> *"The word overheard is that the A3000 is more than just a 68030 dropped into an A2000. Other sources hint at higher-speed custom chips and enhancement of the Amiga's sound capabilities."*

Perhaps the biggest news was the non-news. No major new products were shown. There was no unveiling of the Amiga 3000, although Commodore confirmed that it is in the works. Commodore did say they would be spending a lot more on advertising and marketing—news which was well received by weary developers tired of explaining what computer they work on ("Omega? What's an Omega?").



the MS-DOS File System for the COMMODORE AMIGA

AVAILABLE NOW FOR THE LOW PRICE OF

$30.00 (US)
$36.00 (CDN)

CrossDOS™. . . MS-DOS® DISK ACCESS DONE RIGHT!

The first MS-DOS File System for the Amiga®

- Reads or writes any 360K or 720K MS-DOS or ATARI ST® disks (Version 2.0 or higher) with optional text file filters.
- Transparently accesses MS-DOS files from any utility or application (including file requesters).
- Fully integrates itself into the Amiga operating system.
- Automatically readjusts to different MS-DOS/ATARI ST formats.
- Can be removed after use to reclaim memory.
- Provides an easy installation program.
- Availiable in a READ-ONLY version from the Public Domain or directly from **CONSULTRON** for only **$5.00.**

IN ADDITION to reading and writing any file on an MS-DOS disk, perform the following DOS functions on files and directories.

* Scan any directory
* Create directories
* Rename
* Delete
* Set dates
* Set protection bits
* Seek file positions
* Get disk information
* Add cache buffers

For orders placed through **CONSULTRON** add $3.00 shipping and handling ($8.00 outside the U.S. and Canada). Michigan residents must add the correct sales tax. C.O.D. add $3.00

Send check or money order to:

Please allow up to 2 weeks to process your order.

Dealer inquiries welcome.

CONSULTRON
11280 Parkview
Plymouth, MI 48170
Technical Support
(313) 459 - 7271

Amiga is a registered trademark of Commodore-Amiga, Inc. MS-DOS is a registered trademark of Microsoft, Inc. Atari ST is a registered trademark of Atari, Corp.

Circle 155 on Reader Service card.

Commodore also said they are hiring marketing managers for specific vertical market areas. That's good, because they were significantly understaffed in marketing by all accounts. The plan is to push the 2000 family into vertical markets like publishing and video, and push the 500 into the home and education markets. By hiring people to manage this process and be responsible for it, they have shown that it's more than just lip service this time.

What else is going on with Commodore marketing? Better advertising in vertical market publications, with ads targeted at graphic designers, video producers, and music professionals, and more involvement by Commodore at appropriate trade shows. Taken together, all these moves signify that Commodore is finally marketing the Amiga the way they should.

Commodore is also promising a better effort in selling Amigas to the education market. Better discount structures and more help for educational software developers are among the key elements. Amigas have already become indispensable in the graphic arts departments of some major universities. The challenge now is to get Amigas into K-12 schools.

What's the latest on 1.4? Good news and bad news. The good news is that 1.4 will have plenty of bells and whistles—many conveniences for programmers and users, and a slick-looking Workbench designed by a professional graphic artist. The bad news? If you think it will arrive by Christmas, you must believe in Santa Claus. (Even believing in the Easter Bunny might be too soon.) However, many developers were distressed to learn that the Guru Meditation Error will be changed to something clever and catchy like System Error. Some programmers were muttering about a public domain utility that would bring back the Guru. We'll see.

The astute reader may be asking how compatible the current software applications will be with the new 1.4 software. The official line is that programs that followed all the rules will work just fine, though they may not work with some of the new graphics modes.

Of course, there are many programs that didn't follow the rules, like most games. Another example is DeluxePaint III, which allows you to paint in the overscan border—a no-no under 1.4. Expect many revisions to appear when 1.4 ships sometime in 1990.

Commodore did confirm that they are coming out with the A2630, a 68030 board for the A2000. And they did officially announce that the A3000 exists, but nothing more. The word overheard is that the A3000 is more than just a 68030 dropped into an A2000. Some people say it has been redesigned for higher system throughput, with faster memory chips and faster DMA. Other sources hint at higher-speed custom chips and enhancement of the Amiga's sound capabilities. More mutterings indicate the Amiga could move up to 16-bit graphics, where you could display 256 colors out of 32,768—even a super-HAM mode with all 32,000+ colors available at once.

What about backward compatibility? Some software may need revision. And will there be replacement chip sets for older Amigas? Quite possibly. Commodore should know the value of keeping the installed user base happy. And when will this dream machine be available? Not until 1.4 ships, so don't look for it before 1990. The price has not been set, but the target figure is well under $4,000 list.

Now Commodore plans to provide hardware solutions for the high end. An important element is breaking the 4096 color barrier. They have shown a 256 out of 16.7 million color card. Now they have to ship it and get software support. Is Display Postscript in the Amiga's future? It would be a bold positioning move that would open a lot of eyes, especially in desktop publishing. Of course, you really need powerful graphics coprocessing to make it work, but the Amiga has that (though a faster blitter would help).

In other news, *Commodore* magazine is history. Why was Commodore supporting a magazine anyway? Seems like a silly way for a computer company to waste its time. Maybe it was a good thought when nobody else was doing one, but they should have quit a long time ago. In a somewhat unusual move for Commodore, the staff has been shifted to other jobs rather than being fired.

Speaking of personnel, reports from West Chester indicate that many employees are nervously feeling their necks. If big changes are made, some of the good ol' boys (and gals) may be hitting the road. It's time for everybody to show their worth or show their heels.

The best news of all from DevCon was not official, but it was overheard in more than one spot. As the story goes, Commodore plans to drop the A500 price to perhaps as low as $399 by Christmas time. Of course, this goes hand-in-hand with the plan to put the A500 into the mass-market stores. The Bandito doesn't have any names yet, but Sears would be a good place for an Amiga, don't you think? With that pricing, 1990 might be The Year The Amiga Gun Downs The Bad Guys! A complete system (monitor, computer, 1 meg, 2 drives, and printer) for under $1000—a magic price point at the most magical time of the year.

After all the good news coming out of Commodore, it's almost reassuring to the Bandito to hear a rumor about some truly epic stupidity. West Chester sources say that a prototype of a 6502-based machine that runs at 4 MHz and produces 4096 colors is in the works. And it's 100% incompatible with either the Amiga or the C64.

The Bandito can only hope it's not true, or if it is, that they take the prototype and burn it (or better yet, send it to Atari. It'll never come out then). Doesn't anyone there remember the Plus/4? If you had to dream up the worst possible product for Commodore to introduce, it would sound just like that.

Number crunching

The 1988 figures for computer sales in the US are worth a look. Apple was #1 with 1.271 million units, edging out IBM's 1.229 million units. Who was #3? Commodore, with 665,000 units. That figure includes C64, C128, PC clones, and Amigas. Of course, MS-DOS compatibles, as a whole, are the biggest market of all (about 75% of all PC's sold in the US are MS-DOS machines). Those numbers may change if Commodore can learn how to sell Amigas.

TAKE A BYTE OUT OF HIGH PRICES!

There are fabulous savings on all these goodies

plus a **FREE** mouse pad with orders over $100.

AudioMasterII	$69.95
Aunt Arctic Adventure	25.95
Battle Chess	34.95
Battle Tech	36.95
Baud Bandit	32.95
Blood Money	29.95
Deluxe Music 2.0	72.95
Deluxe Paint III	109.95
DigiPaint III	69.95
Double Dragon	29.95
Dragon's Lair	41.95
Dungeon Master	27.95
FA/18 Interceptor	37.95
Falcon	36.95
Kind Words	59.95
Licence To Kill	29.95
Lords of the Rising Sun	34.95
OutRun	34.95
Phasar	63.95
Pioneer Plague	28.95
Publisher Plus	49.95
Rocket Ranger	35.95
Rush N' Attack	29.95
Sinbad & Throne of Falcon	19.95
SimCity	31.95
Sonix	56.95
Spellbound	28.95
Super Scramble	29.95
Sword of Sodan	35.95
Test Drive II	29.95
The Three Stooges	35.95
Total Eclipse	29.95
Triad	29.95
TV Sports Football	34.95
V.I.P. Virus Protection	29.95
Where in World is Carmen	32.95
Who Framed Roger Rabbit	32.95
Who! What! When! Where!	65.95
Enhancer Amiga DOS 1.3	20.95
Accessories & Hardware	
Epyx Joystick	14.95
ErgoStick Joystick	19.95
Internal 3½" Floppy Drive for 2000	99.95
California 3.5" drive	139.95
Supra 2400 Baud Modem	129.95
Spirit SC 501 ½ Meg. Trapdoor internal expansion with clock/calendar for Amiga 500. 0K	63.95

plus shipping and handling

Don't see it here? We've probably got it. Call us.

B&B COMPUTER

Your Amiga Source

P O Box 575719
Murray, Utah 84157-5719
1 800 347 8004

VISA MasterCard

Circle 110 on Reader Service card.

The rising value of the dollar changed profit into loss for Commodore during the last quarter. Since three-quarters of their sales are overseas, the exchange rate really hammered their profit picture. The pressure is building to make good sales numbers for the rest of the year. If Europe slows down, then the US division has to pick up the slack. (A baptism of fire for Mr. Copperman.)

Even though nobody asked, the Bandito thinks Commodore could make a big splash by being the first computer manufacturer to offer a CD-ROM drive built into a computer. Offering a low-cost CD-ROM option for the A500 (or other models) would also be dynamite. How about a retail price of under $500? It's possible. Of course, it would also have to play CD audio disks. C'mon Commodore, get with it. Even IBM is talking about CD-ROM and multimedia these days.

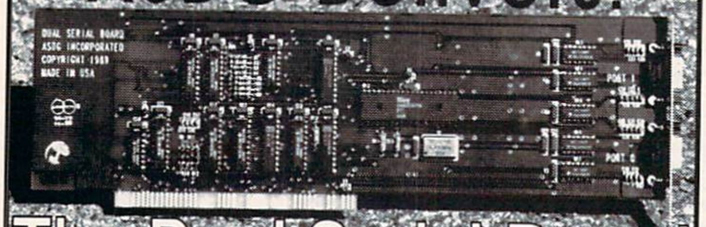
Speaking of multimedia (or as the Bandito thinks of it, "desktop buzzwords"), IBM's "Home PC" is heading our way in 1991. The specs sound hot: 386SX processor, DVI chipset for near real-time video, a digital signal processor for great sound and built-in 9600 baud modem, MS Windows and a mouse, and a built-in CD-ROM drive. The price will probably be in the $2500 to $3000 range, so they may be slanting it more towards the home office rather than the home entertainment market. It's certainly a target for Commodore to take into consideration. The Amiga should have all of those features by then, and with a better installed base, too.

Commodore hopes to capture the interactive video market with the Amiga. Their first target is interactive point-of-sale video systems—basically, a laserdisc player hooked up to an Amiga inside a kiosk, where customers can go up and select items they want to see. The Amiga's advantages are NTSC output compatibility, making genlocking cheap. The entire Amiga interactive video system would go for under $3000 with laser disc player and genlock.

A comparable IBM-type system would run at least $1000 more, and a

Circle 160 on Reader Service card.

Others Make Claims, ASDG Delivers.



The Dual Serial Board

Fast, Compatible, Complete!

● Two high speed RS-232C serial ports for the Amiga 2000. ● IBM PC AT style (full 7 wire plus RI and DCD) connector for each port. ● Send and Receive at 110 to 115200 baud. ● Complete Amiga system software (both Exec *and* DOS). ● Using our exclusive Serial DisPatcher, *any* program which uses "serial.device" can use any port *without* patches. ● Public domain and shareware software on the distribution diskette. ● One year warranty.

ASK YOUR DEALER!

ASDG
Incorporated

925 Stewart Street
Madison, WI. 53713
(608) 273 - 6585

Amiga is a trademark of Commodore-Amiga, Inc. IBM PC AT is a trademark of IBM Corp.

Macintosh system similarly configured would be around $7000. Commodore estimates this to be a billion-dollar-a-year market in training, sales, and education. They are working on an authoring system to make it easier to create interactive videos. No word yet on whether this technology might reach the average user.

Developments and non-developments

Musicians take note: MasterTracks Pro and the rest of Passport's music software line is currently being ported to the Amiga; expect software to arrive this fall. More and more music software publishers are developing for the Amiga as the machine is increasingly being accepted as a serious music machine. One important reason is the Amiga's video capabilities, which are more important to music videos.

While we're talking about music, how about a new IFF format for sampled sounds without the incredible limitations of 8SVX? A standard is a must, but let's develop one that really takes advantage of the Amiga's sound capabilities. Let's not let an outmoded file standard hold us back from getting the best sound possible from our favorite computer.

DeluxeVideo III may be the last Amiga creativity product from Electronic Arts. [*Editor's note: Electronic Arts insists that it "is and will continue to be dedicated to its Deluxe line of products..."*]

The word is that Electronic Arts has killed plans for Instant Music II. [*Editor's note: Electronic Arts would not comment on Instant Music II, as it is an unannounced product.*] Most of the people involved with producing creativity products have been reassigned, transferred, or otherwise departed. Seems like they want to concentrate on games, which is really where the company has always put most of its effort.

Interestingly, both Epyx and Mediagenic have had great trouble making their creativity products divisions profitable, and they're cutting back in those areas as well. The Bandito figures that games and work-type software just don't mix.

The Bandito is considering regularly setting aside a part of the column for legal issues, since they're so common these days. The latest: Xerox has prevailed upon Metaphor to pay a licensing fee for using windows, icons, and a mouse in their interface that IBM has licensed. Xerox may demand licensing fees from everybody using a windowing interface. That means Amigas, and maybe even GEOS. What it really means is good news for lawyers, who stand to make a lot of money. The whole thing seems silly to the Bandito. Xerox has never copied anything, have they?

Hottest items in the stores: the A-Max Macintosh emulator from ReadySoft. Also the 1 meg Agnuses are still in very short supply, but all A2500's and 2000's are now shipping with them installed. The situation may have cleared up by the time you read this. Certainly the dealers hope so. At about $150 each, they're doing a good business in upgrades. On the software scene, PageRender 3D looks hot. The Bandito sees more 3D software on the Amiga than on the Mac and the IBM combined. And the results are far better on the Amiga.

If you haven't noticed lately, RAM prices are dropping faster than Mediagenic's profits. With the new 4 megabit chips being introduced, prices are sure to be driven even lower. The Bandito expects a megabyte of memory to be under $100 by the end of the year, if not before. Good news for animation fans.

The Bandito's colleague in the Macintosh market keeps hearing that Apple plans to breathe life into the corpse of the IIGS by including its chip set in the new low-cost Macintosh under development for next year. Wishful thinking.

Adding those chips would bring the cost up too much. Apparently Apple just wants to keep the schools happy by giving them an upgrade path for their old software. More believable is some sort of add-in card that gives IIGS compatibility. Either way, an Amiga 500 under $1000 will make the Cheap Mac look pretty sick. Another crab apple, if you ask the Bandito.

•AC•

SNAPSHOT

Four Amiga Game Reviews

by R. Bradley Andrews

BattleTech

An era is at an end. Many computer users are familiar with Infocom, the company that produced a myriad of text adventures for many different computers. While their adventures were rich and detailed in their time, text adventures have become much less popular with the advent of more powerful computers with more graphics capabilities. Unable to generate sufficient profits, Infocom has been bought out by Mediagenic (formerly Activision), who reportedly closed their offices. So ends an era.

Fortunately, all is not lost. Under the Infocom label, Mediagenic has just released a new, graphics-based role-playing game (RPG) for your gaming pleasure—BattleTech: The Crescent Hawk's Inception. It is the first in a possible series of computer games based on a board game produced by FASA. The backdrop for the game is futuristic robotic combat, as popularized on television, in comics, and in board games.

You begin the game controlling the character Jason Youngblood, the son of an illustrious family whose origins can be traced back more than 1,000 years to the early 21st century. Since your father is Captain of Security Forces for the planet, you have added pressure to successfully overcome the obstacles put in your way.

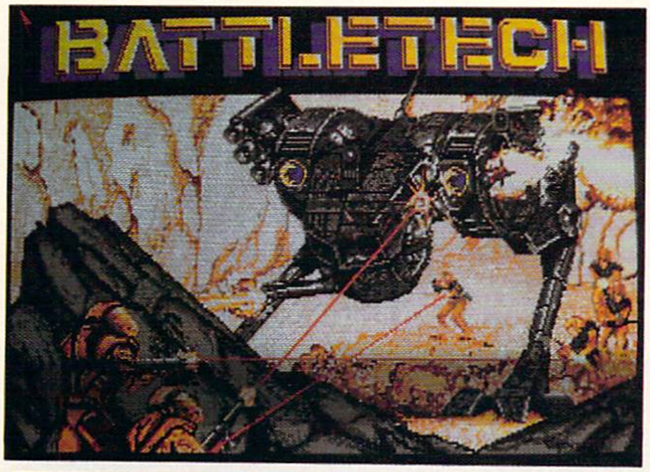


Figure One: *The opening screen to BattleTech*

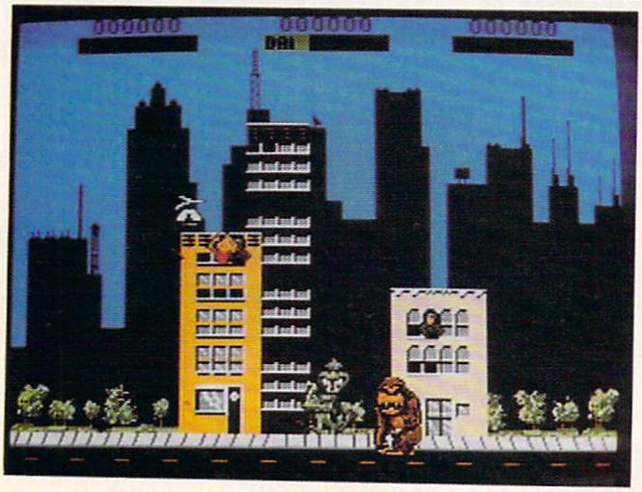


Figure Two: *Lizzie the Lizard and George the Big Ape stalk the city streets in Rampage*

Play begins with your MechWarrior training in the training complex on the planet Pacifica. This training, while simple at first, becomes increasingly grueling. It is designed to refine your skills completely before you are finally put in charge of one of the galaxy's most potent fighting machines, the mighty Mech. The technology to build new Mechs has been lost, so the existing ones are guarded with extreme care, since they hold the key to swaying the battles fought in occupied space.

In addition to your Mech training, you have other opportunities to sharpen your personal skills—at a price. Fortunately, money is periodically "sent from home" and deposited in your account. By budgeting properly and playing the stock market successfully, you can make these small sums blossom quickly into sizable amounts. You should maximize all your skills while you're in the training city, since once the entire planet has been opened to you, only your medical and repair skills can be increased.

A good strategy for multiplying your money in the stock market is to put all of it in the high-risk BakPhar company. Save the game before investing, so if you lose money you can restart and try again. If you gain money, re-save and continue the process over. I have found that the optimal amount to invest in BakPhar is about 8000 C-Bills; more always seems to lose. Though you can invest your money only a limited number of times while in the training area, once in the spaceport, you can invest as much and as often as you'd like. The only limiting factors are your patience and willingness to watch your money grow in little chunks.

Unfortunately, your preparation is interrupted before you can finish completely. If you can maneuver successfully, you will find yourself wandering around the planet's surface, searching for the starport and a way out of trouble. While there, you will come into contact with your new team, the Crescent Hawks. After gathering all the members, you need to fix a damaged holodisk and then locate a hidden weapons cache—which you will turn over to the leader of your side of the conflict. Once these tasks are completed, your quest will be over.

The entire game is menu-driven and features very nice looking graphics with a smoothly scrolling background. The sound is adequate and complements game play. Although you can play the entire game from the mouse, I found that the keyboard worked better, since either the keypad or the arrow keys could be used to move on-screen characters.

While only a limited number of actual options are available during play, game play is fairly fluid and not too limited. Games can be saved or loaded at any time, though only six slots are available for doing so. Frequent saving is important, but the disk-access delays can be a bit annoying. Hard disk users should have fewer problems in this area since the game can be loaded onto and played from a hard disk.

Despite its nice points, BattleTech does have several flaws that mar its otherwise excellent exterior. First, the advertised theme of the game, robotic combat, plays a small role during play. Sure, you participate in a few training missions and some battles on the planet's surface, but the initial thrill of combat soon wears off. I had the computer perform most combat to avoid the time delay associated with its resolution.

Second, I found the ending of the game to be extremely anti-climactic. The closing speech announces that the universe is now open to your group, but the game ends before you can explore it.

Third, it is possible for key members of your team to be traitors. This possibility would not be so bad were it not for the fact that you need all your team members later in the game. It is advisable to save the game before picking up new team members, since traitor status is apparently decided randomly at the time you pick them up. The first combat should flush out any traitors. If you do pick up a traitor, restart the game at the point before you picked him up. Otherwise, you will not be able to finish the game.

Fourth, the segment in which you unlock the arms cache is rather tedious—a lot of busywork that doesn't really add to the quality of play. And finally, a lot of playing time is spent simply roaming around the planet's surface from one spot to another. It would have been useful to have a way of automatically sending the team to a desired spot.

I would rate BattleTech as a reasonable entry-level adventure, noting the above reservations. I actually did enjoy my playing time, though I do regret that the ending did not live up to the level of effort I invested to reach it. If your RPG tastes are on the sophisticated end of the spectrum, you might want to avoid this one. Given the relatively few competitors in BattleTech's genre, however, it may be your only choice.

Rampage

Next on the list this month is Rampage from Mediagenic (under their Activision label). This game is an adaptation of the coin-op version notable for its mayhem and destruction. While the game does have a high-score feature, your primary goal in playing it is simply to destroy as many cities as possible before your energy level runs out.

Three monsters are available for use during play: Ralph the Wolf, who has the meanest punch; George the Big Ape, who's the best at scaling buildings; and Lizzie the Lizard, who's the fastest thing on four scaly feet. Originally simple, ordinary human beings, they all came in contact with a strange chemical substance that made them the monstrosities they are today.

You destroy buildings by punching out their infrastructures until they come crashing down. Effective demolition involves punching both sides of a building, which requires you to use your creature's scaling ability. If you're adept, you can even jump from one building to another, accelerating the process.

Most playing time is spent on the city screen, which displays a side view of the buildings you are attempting to flatten. Once one city has been cleared (or "renovated," depending upon your perspective) it's off to the next for more fun. Between cities, news flashes appear, chronicling the adventures of your monsters as they blaze a trail of destruction across the U.S. Be warned: These messages make liberal use of puns and other forms of "humor."

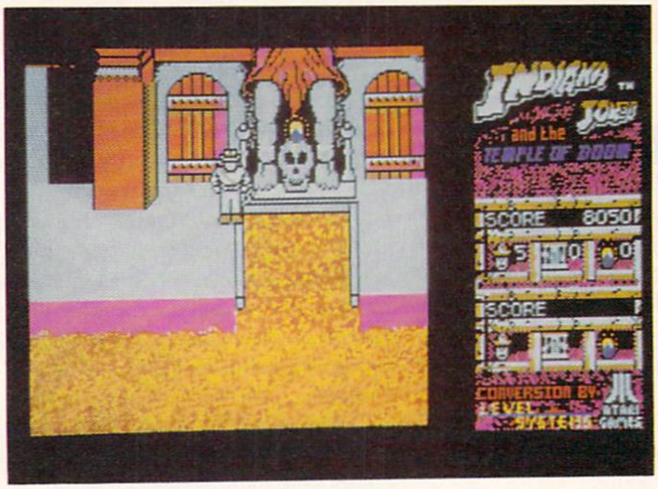


Figure Three:
You play the whip-wielding Indy in Indiana Jones and the Temple of Doom

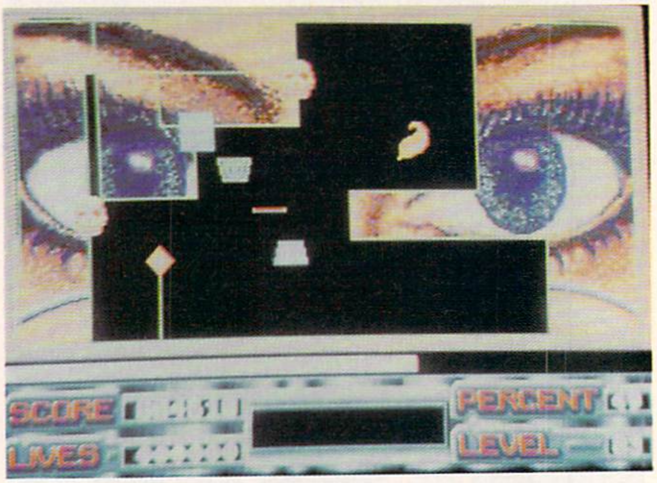


Figure Four:
Use your power crystal to cut out the screen and expose the underlying picture in Powerstyx

The chief hindrance to your cross-country rampage is your ever dwindling energy supply; when it runs out, your monster shrinks back to a "normal" human form and beats a hasty retreat off the screen. Soldiers, tanks, and helicopters are constantly firing their arsenals at you, speeding up your energy loss and bringing about your demise. Fortunately, eating humans or any of the other "goodies" that are uncovered when you demolish a building can restore some of your lost strength, prolonging your life. However, you must be careful not to put everything you see in your mouth. Some things are harmful to your health, such as active light bulbs, which can really send some "bad vibes" through your system.

The graphics are acceptable, but as you might expect, they do fall short of those in the arcade version. Some of the objects are somewhat block-like and difficult to recognize. I'm not certain, but I think that Mediagenic could have done a little better job in this area.

All three monsters can play simultaneously, controlled by joystick, mouse, or keyboard. Since the active controls can be changed, nearly any possible configuration is possible. One drawback, however, concerns my favorite controller for this type of game: the joystick. Both jumping and punching are controlled by the fire button. This configuration can often cause you to perform the wrong action. The programmers aren't necessarily at fault here: What we really need is a two-button joystick.

Rampage is a reasonable adaptation of the arcade game, allowing you to destroy major urban centers from the comfort of your own home. The Amiga version, however, is missing some of the spark that made its coin-op progenitor so enthralling. For this reason, I highly recommend that you try before you buy.

Indiana Jones and the Temple of Doom

Next up is Indiana Jones and the Temple of Doom, another adaptation of an arcade game—this time from Mindscape. As you may have guessed, this is the "official" game based on the second movie in the Indiana Jones series. In it, you guide the whip-wielding Indy through an attempt to rescue the enslaved children of Mayapore, recover the stolen Sankara Stones, and put an end to the evil reign of the ruthless high priest Mola Ram. While a whip may not seem like much, Indy's proficiency with it makes it more than enough for the task at hand.

Each segment of the game is played in one of three different areas: the Mine Caverns, the Mine Tunnels, or the Temple of Doom itself. In the Caverns, Indy must work his way up a complex arrangement of ladders and ledges to reach the entrance to the Tunnels. This first area is also where the captive children are imprisoned. Indy's whip will make short work of their bonds, but each is held in a different area, and it may require some hunting to find them all.

After the Caverns are completed, Indy jumps into the mine car for the ride of his life. The Tunnels are basically a maze of rail-car tracks which has to be successfully navigated all the way to the bottom. Some tracks end abruptly, sending our hero plummeting into the darkness. Once you learn the pattern for each track, the Tunnels are relatively easy to complete. (For the safest route, keep to the right.) Speed is also important. If Indy takes too long, he will encounter a massive fire at the end of the tracks and lose another life.

At the end of the Tunnels lies the Temple itself. This is by far the easiest area, as Indy must simply maneuver his way around the fire pit, stunning any guards that get in his way. Then, once he reaches the Sankara Stone, he can steal it and enter a new cavern to free even more children.

After this, Indy must find and cross a rope bridge for a face-to-face encounter with Mola Ram himself. If Indy defeats the high priest, he then enters a bonus round to rack up as many points as possible. Unfortunately, this bonus round seems to end only when Indy runs out of lives. As a result, intentional suicide appears to be the only way out of this phase.

The game features a total of three Caverns-Tunnels-Temple sequences, each of which can be played at one of three difficulty levels: easy, medium, or hard. The two higher levels feature spikes that pop up at various locations in the caverns, as well as more aggressive guards in the Temple.

The graphics in the game are very crisp and clear, comparing favorably with those in the coin-op version. The joystick is used to control Indy during play and generally works well. Sometimes problems do arise because of imprecise positioning. For example, I have had guards walk right up to Indy and knock him out while he flailed away with his whip. It can also be difficult to align Indy with ladders, often resulting in nasty falls that cost a life.

The game is a good rendition of the "maneuver, kill, and free the prisoners" arcade theme. Once you master the mechanics of it, however, the limited number of screens do not provide much of a challenge. While The Temple of Doom may be a worthwhile acquisition for some, the infrequent buyer should probably look into some of the better action games available on the Amiga.

Powerstyx

Finally this month there is Powerstyx. Based on the arcade game Qix, it goes a step further, resulting in a game that is both enthralling and slightly bizarre. Programmed by Magic Bytes and distributed by DigiTek Software, it follows the somewhat off-the-wall theme begun with their other release, Hole-In-One Golf.

The goal in Powerstyx is to use your power crystal (controlled with the joystick) to cut out sections of the screen, exposing the underlying picture. When a sufficient amount of the picture is exposed, the level is over, the entire picture is displayed, and the next level is loaded.

Of course, there are several enemies provided to complicate this task. One is a kind of energy monster who randomly zips around the interior of the unexposed area. While it's pretty to look at, don't touch it! Any contact results in immediate death. Other monsters—equally fatal when touched—roam around the edges, but these move at a fixed speed and can be temporarily stunned if surrounded with a newly exposed area.

Several objects that can aid you in your task periodically float across the screen. The heart gives an extra life, the tube gives more speed, the clock gives more time, moneybags add to your point total, and the door immediately transports you to the next level. The cross is to be avoided, however: it will end your current life if you catch it. Sometimes a question mark will float across the screen. The only way to find out which object it represents is to catch it—and hope for the best.

Letters also appear periodically in

the playing area. As each is captured, it is placed in the word-holding area. When the entire word describing the current picture is spelled out, you are advanced to the nex level.

The graphics in the game are well done and match the theme quite well. The background pictures are also drawn nicely, albeit with a rather bizarre motif. Some of the more interesting designs include a half-plane, a half-bird, and a two-legged alien eye—definitely something to see. Uncovering new pictures does add something to the quality of play, and can keep you coming back even after the game mechanics are no longer challenging. This is not to say that the game ever becomes easy. The speed of the opponents gradually increases with each level, adding more and more complexity.

Powerstyx is probably the best of the three arcade games discussed this month. It's a game that's sure to be enjoyed by all who purchase it.

•AC•

Mediagenic/Infocom
BattleTech: The Crescent Hawk's Inception
3885 Bohannon Drive
Menlo Park, CA 94025
(415) 329-0800
Price: $49.95
Inquiry # 256

Mediagenic/Activision
Rampage
3885 Bohannon Drive
Menlo Park, CA 94025
(415) 329-0800
Price: $39.95
Inquiry # 257

Mindscape
Indiana Jones and The Temple of Doom
3444 Dundee Road
Northbrook, IL 60062
(312) 480-7667
Price: $49.95
Inquiry # 258

DigiTek Software
Powerstyx
104 West Seneca, Suite 4
Tampa, FL 33612
(813) 933-8023
Price: $34.95
Inquiry # 259

Be an Amazing Writer!

Get paid for what you love to do...
using your Amiga!

Contact *AC* for a Writer's Guide.

("Palettes", continued from page 49)

Listing One

```
'* * * * * * * * * * * * * * * * * * *
'*            LISTING 1                *
'*  a simple example of dithering      *
'*            source code              *
'*                by                   *
'*          Robert D'Asto              *
'* * * * * * * * * * * * * * * * * * *

DIM P%(1)

PALETTE 3,1,0,0

LINE (100,50)-(200,150),1,b
LINE (250,50)-(350,150),1,b
LINE (400,50)-(500,150),1,b

PAINT (1,1),2,1
PAINT (101,51),3,1

P%(0)=&HA5A5
P%(1)=&H5A5A

PATTERN ,P%

PAINT (401,51),3,1

P%(0)=&HFFFF
P%(1)=&HFFFF
PATTERN ,P%
```

Listing Two

```
'* * * * * * * * * * * * * * * * * * *
'*            LISTING 2                *
'*  produces 136 apparent colors       *
'*      on a 16-color screen           *
'*           source code               *
'*               by                    *
'*          Robert D'Asto              *
'* * * * * * * * * * * * * * * * * * *

SCREEN 1,640,200,4,2
WINDOW 2,,(0,0)-(631,186),0,1

DEFINT a-z

DIM PatArray(1)

PALETTE 4,.6,0,0            'starting colors
PALETTE 5,1,1,.13
PALETTE 7,.4,.4,.4
PALETTE 10,.7,.7,.7
PALETTE 11,.3,.3,0
PALETTE 12,1,0,0
PALETTE 13,0,.6,0
PALETTE 14,0,0,1
PALETTE 15,1,0,1

FOR x=0 TO 629 STEP 37  'create display
  LINE (x,0)-(x,184),2  'grid
NEXT

FOR y=0 TO 184 STEP 23
  LINE (0,y)-(629,y),2
NEXT

FOR x=1 TO 556 STEP 37   'display starting
  PAINT (x,1),c,2        'colors
  c=c+1
NEXT

PatArray(0)=&HAAAA         'create bit
PatArray(1)=&H5555         'pattern
PATTERN ,PatArray

DitherFill 593,1,0,1       'start dithering

x=1:y=24:c1=0:c2=2
WHILE count <= 118
  ColorCycle c1,c2
  DitherFill x,y,c1,c2
  x=x+37:IF x > 593 THEN x=1
  y=y+23:IF y > 162 THEN y=24
  count=count+1
WEND

LOCATE 23,21:COLOR 1,2     'brag a little
PRINT "136 colors on a 4-bitplane screen!"

PatArray(0)=&HFFFF         'reset fill
PatArray(1)=&HFFFF         'PATTERN
PATTERN ,PatArray

END

SUB DitherFill (x,y,Clr1,Clr2) STATIC
  COLOR Clr1,Clr2
  PAINT(x,y),Clr1,2
END SUB

SUB ColorCycle(c1,c2) STATIC
  c2=c2+1
  IF c2 > 15 THEN
    c1=c1+1
    IF c1 >= 15 THEN c1=0
    c2=c1+1
  END IF
  IF c2 > 15 THEN c2=0
END SUB
```

•AC•

Bug Bytes

The Bugs and Upgrades column

by John Steiner

A-Max, the Macintosh emulator for the Amiga, is a unique product that addresses some Amiga users' desire to run Macintosh software on their Amigas. A revision to the program has been made that corrects a problem with the Setpatch command found in version 34.25 Workbench systems (the latest available Workbench).

The original A-Max fails to operate floppy drives properly if the new Setpatch command has already been issued. An upgrade is available which fixes not only this problem, but several other minor bugs, including a problem with the print routines.

To determine whether or not you have the latest version, simply check the Readme file on the master disk. If the Readme file mentions HyperCard, you have the latest version. If you do not have the latest version, you can send ReadySoft your original disk, and they will replace it for you by mail.

* * *

As reported last month, the Fatter Agnus chip is being installed in earlier model A2000's and A500's around the country. The major reason for the upgrade is to be able to run Professional Page and Professional Draw simultaneously.

After performing the necessary surgery on my own machine, I immediately started work on a newsletter. But before long, I encountered all kinds of problems when I ran the two programs together. It turns out that the problem was related to a specific document (my empty newsletter template), not to the upgrade. Once I corrected the problem, the two programs worked together very nicely.

In the process of tracking down that problem, however, I discovered that early Fatter Agnus chips could have a defect. There is a simple test you can perform to determine whether there is a problem with a specific Fatter Agnus chip. From your original Workbench disk, run the Clock and Lines programs simultaneously in the Demos drawer. If, after letting the two programs run for about a half-hour, you find that the Clock program behaves in an unusual fashion, or that the Lines program starts drawing lines outside of its defined window, you have a defective Fatter Agnus chip. Commodore will replace them under warranty via an authorized service center.

* * *

There is an ARexx update available from William Hawes. If you did not request notification from ARexx—either by postcard or by their automatic upgrade policy—when you registered your software, you can now request an update to version 1.10. It costs $5.00, and can be obtained by contacting Mr. Hawes (see address at end of column).

* * *

I received several letters this month from readers with bug reports. Jacques Chatenay of Fargo, ND reports a few bugs in Superbase Professional. Both problems have been worked through by technical support at Precision Software.

First, there is a problem with some kind of contention between the PRINT and LABEL commands. If you use the PRINT command for printing text, printer codes, or even just a blank PRINT command before using the LABEL command, the LABEL command will not work correctly. The symptom is that the appropriate linefeeds are not issued, causing each field in the label to be printed on the same line.

To work around this problem, instead of using the PRINT command, which outputs to the printer, use Set Printer On followed by the ? character, which outputs to the current output device. This allows the LABEL command to operate properly.

The second problem in Superbase is that the forms editor generates incorrect code in a report program when you attempt to use the SUM function to subtotal a group or total a report.

To demonstrate the problem—and how to fix the incorrect code—I will use a fictitious file named CLIENTS, which has two fields, Salesperson (a text field) and Totals (a numeric field). Open the file and select Group from the Report menu. You will get a Select Group dialog box containing a list of fields. Select Salesperson to group on and click on OK. Three boxes will appear on the form, labeled "Before group Salesperson," "Select," and "After group Salesperson." From the Set menu, choose Field, click in the Select box, enter the field you want subtotaled (Totals, in our example), and then click on OK. Select Function from the Set menu, click in the "After group" box, and select SUM from the requester. Then click on the Totals field chosen earlier.

To have the program total the report, select Report from the Report menu, adding two more boxes—"Before report" and "After report"—to the screen. Still using the function tool from the Set menu, click in the "After report" box and select Sum and Totals in the same manner you did earlier.

Now save your completed form. When you select Open Form from the Project menu in Superbase, answer the requesters as you normally would. Eventually, an error message reading "Can't Do This, Invalid Parameter" will appear and the report will stop. Click on OK to enter the program editor. The cursor will be on the line where the program failed, but the error is actually before that line. Look above it for a line that begins with AFTER GROUP Salesperson.CLIENTS. Then look further up for a line that reads GROUP, followed by the same fieldnames. At the end of the GROUP command, add a comma, and then the fieldname of the field that is being summed. For example:

```
GROUP Salesperson.CLIENTS,Totals
```

Next, find the line that contains REPORT. After the word REPORT, leave a space and then enter the field being totaled. For example:

```
REPORT Totals
```

Now run the program again. It should work properly.

* * *

Jim McCabe of Upland, CA reports on a couple of problems with Maxi Plan 1.8g and 1.9b. Version 1.8g crashes frequently when he attempts to make charts, requiring him to reboot. The program has also crashed randomly when he scrolls through a large spreadsheet with the screen arrow gadgets.

Version 1.9b no longer exhibits the problem with the chart functions, but it still has a problem with random crashing when he scrolls through the spreadsheet. Jim is also having problems printing a chart larger than a quarter of the size of a 8.5 x 11-inch page. He also reports that the program redraws a chart if the right mouse button is pressed, causing a delay while the screen is redrawn. It seems that Maxi Plan spends too much time doing screen redraws.

Now that Maxi Plan has been upgraded to Plan/It, as reported in an earlier *Bug Bytes*, some of these problems have been fixed. One problem with the new Plan/It which I discovered relates to saving IFF picture files. The graphs saved in low-resolution 320 x 200 mode are just fine, but graphs saved in 640 x 400 mode all contain a black box in the center of the image where the file requester box is located just prior to the IFF file creation. Since the resulting IFF graph is useless, Plan/It should redraw the graph once more, just before it writes the graph to disk.

* * *

Ron Battle of Albuquerque, NM writes to ask if there is an update to allow Music Studio from Activision to run under Workbench 1.3. His attempts to contact Activision have yielded no results. I do not have an address on file for them. If anyone knows how to contact these people (assuming they are still in the Amiga software business), please pass along the information.

Mark Hellman of Wheatridge, CO writes with a bug report for version 3.11 of B.A.D. He accidentally discovered that B.A.D. locks up the Amiga irretrievably when it encounters two disks with the same name. There is no Guru message; the computer simply locks up. The problem is easily solved: Before running B.A.D., simply be sure that no two disks have the same name. If some do, use the RENAME function in either AmigaDOS, CLI or the Workbench to change one (or both) of the names

* * *

I received an upgrade notice from some very pleasant folks at Blue Ribbon Bakery, publishers of Who! What! When! Where!, an appointment calendar and scheduling program. The list of improvements to the latest version (1.3) includes an ARexx port, a new "Kill Timer" program, printing to disk, faster disk I/O, the ability to repeat appointments on a certain day of the week every month, and several other enhancements. To upgrade your program, send in your original disk and a check or money order for $10.00 payable to Blue Ribbon Bakery.

* * *

According to the people at Pixelations, version 1.1 of PrintScript is ready. Actually, PrintScript has been renamed PixelScript because of a trademark conflict. The 1.1 version has an Intuition-style user interface that allows you to start from an icon and use a file requester (without dealing with the CLI). The font-loading code is now roughly ten times faster for our fonts, and about two times faster for other fonts. This improvement doesn't make pages print that much faster, but the startup time is reduced significantly.

Memory usage is also improved. Curves in graphics are much faster and require much less memory. Support for both WordPerfect .ps files and City Desk 2.0 is also in this version. PageStream EPS import can be handled, so you can use PageStream to import EPS and print it with PixelScript. You can also add Pixelscript to the list of many programs that now have added ARexx support. The program's price has been increased to $149.00 list. More fonts will be available soon. Registered owners of version 1.0 can call or write for details on upgrading to Pixelscript 1.1.

* * *

That's all for this month. If you have any workarounds or bugs to report, or if you know of any upgrades to commercial software, please notify me by writing to:

John Steiner
c/o Amazing Computing
Box 869
Fall River, MA 02722

...or by leaving EMail to:

Publisher on People Link
or
73075,1735 on CompuServe.

•AC•

William S. Hawes
P.O. Box 308
Maynard, MA 01754
(508) 568-8695
(Inquiry #200)

Blue Ribbon Bakery
1248 Clairmont Road, Suite 3D
Atlanta, GA 30030
(404) 377-1514
(Inquiry #201)

Pixelations, Inc.
Box 547
Northboro, MA 01532
(508) 393-7866
(Inquiry #202)

ReadySoft Corporation
30 Wertheim Court Unit 2
Richmond Hill, Ontario, CN L4B 1B9
(416) 731-4175
(Inquiry #203)

Precision Software
8404 Sterling St., Suite A
Irving, TX 75063
(214) 929-4888
(Inquiry #204)

Amazing COMPUTING™
Your Original AMIGA® Monthly Resource
For The Commodore AMIGA®

Documented Evidence

Amazing Computing has been providing solid support for the Commodore-Amiga for over 3 years with helpful, informative issues and we have the documents to prove it!

Since February 1986, Amazing Computing™ has been providing users with complete information for their Amigas. This store house of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga any user will find useful. AC was the first magazine to document CLI, tell its readers how to connect a 5 1/4 IBM drive, describe a 1 meg upgrade hardware project for the A1000, and many more. And AC is doing more everyday to help Amiga users be more productive and entertained.

If doing more with your Amiga is what you want, subscribe to AC today and if you do not have a full set of back issues, order soon. Unfortunately, nothing lasts forever, and the availability of some of our Back Issues is definitely limited. Complete your Amazing Computing™ library today, while these issues are still available, by completing the order form in the back of this issue.

Back Issues are $5.00 US, $6.00 Canada and Mexico, $7.00 Foreign Surface
All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.
Visa and MasterCard accepted on orders of $20 or more.

1-800-345-3360

Volume 1 Number 1 Premiere 1986

Super Spheres By Kelly Kauffman An ABasic Graphics prog.
Date Virus By J Foust A disease may attack your Amiga!
EZ-Term by Kelly Kauffman An ABasic Terminal program
Miga Mania by P. Kivolowitz Programming fixes & mouse care
Inside CLI by G. Musser a guided insight into the AmigaDos™
CLI Summary by G. Musser Jr. A list of CLI commands
AmigaForum by B. Lubkin Visit Compuserve's Amiga SIG
Commodore Amiga Development Program by D. Hicks
Amiga Products A listing of present and expected products

Volume 1 Number 2 March 1986

Electronic Arts Comes Through A review of software from EA
Inside CLI: part two G. Musser Investigates CLI & ED
A Summary of ED Commands
Live! by Rich Miner A review of the Beta version of Live!
Online and the CTS Fabite 2424 ADH Modem by J. Foust
Superterm V 1.0 By K. Kauffman A term. prog. in Amiga Basic
A Workbench "More" Program by Rick Wirch
Amiga BBS numbers

Volume 1 Number 3 April 1986

Analyze! a review by Ernest Viverios
Reviews of Racter, Barataccas and Mindshadow
Forth! The first of our on-going tutorial
Deluxe Draw!! by R. Wirch An Amiga Basic art program
Amiga Basic, A beginners tutorial
Inside CLI: part 3 by George Musser George gives us PIPE

Volume 1 Number 4 May 1986

SkyFox and Articfox Reviewed
Build your own 5 1/4 Drive Connector By Ernest Viveiros
Amiga Basic Tips by Rich Wirch
Scrimper Part One by P. Kivolowitz prog to print Amiga screen
Microsoft CD ROM Conference by Jim O'Keane
Amiga BBS Numbers

Volume 1 Number 5 1986

The HSI to RGB Conversion Tool
by S. Pietrowicz Color manipulation in BASIC
AmigaNotes by Rick Rae The first of the Amiga music columns
Sidecar A First Look by John Foust A first "under the hood"
John Foust Talks with R. J. Mical at COMDEX™
How does Sidecar affect the Transformer
an interview with Douglas Wyman of Simile
The Commodore Layoffs by J. Foust A look Commodore "cuts"
Scrimper Part Two by Perry Kivolowitz
Marauder reviewed by Rick Wirch
Building Tools by Daniel Kary

Volume 1 Number 6 1986

Temple of Apshai Triology reviewd by Stephen Pietrowicz
The Halley Project: A Mission reviewed by S Pietrowicz
Flow: reviewed by Erv Bobo
Textcraft Plus a First Look by Joe Lowery
How to start your own Amiga User Group by William Simpson
Amiga User Groups
Mailing List by Kelly Kauffman a basic mail list program
Pointer Image Editor by Stephen Pietrowicz
Scrimper: part three by Perry Kivolowitz
Fun With the Amiga Disk Controller by Thom Sterling
Optimize Your AmigaBasic Programs for Speed by Pietrowicz

Volume 1 Number 7 1986

Aegis Draw: CAD comes to the Amiga by Kelly Adams
Try 3D by Jim Meadows an introduction to 3D graphics
Aegis Images/ Animator: a review by Erv Bobo
Deluxe Video Construction Set reviewed by Joe Lowery
Window requesters in Amiga Basic by Steve Michel
ROT by Colin French a 3D graphics editor
"I C What I Think" Ron Peterson with a few C graphic progs
Your Menu Sir! by B Catley program Amiga Basic menues
IFF Brush to AmigaBasic 'BOB' Basic editor by M Swinger
Linking C Programs with Assembler Routines...by Gerald Hull

Volume 1 Number 8 1986

The University Amiga By G.Gamble Amiga at Washington State
MicroEd a look at a one man army for the Amiga
MicroEd, The Lewis and Clark Expedition reviewed Frizelle
Scribble Version 2.0 a review
Computers in the Classroom by Robert Frizelle
Two for Study by Frizelle Discovery & TheTalking Coloring Book
True Basic reviewed by Brad Grier
Using your printer with the Amiga
Marble Madness reviewed by Stephen Pietrowicz
Using Fonts from AmigaBasic by Tim Jones
Screen SaVer by P. Kivolowitz A monitor protection prog. in C
Lattice MAKE Utility reviewed by Scott P. Evernden
A Tale of Three EMACS by Steve Poling
.bmap File Reader in Amiga Basic by T Jones

Volume 1 Number 9 1986

Instant Music Reviewed by Steve Pietrowicz
Mindwalker Reviewed by Richard Knepper
The Alegra Memory Board Reviewed by Rich Wirch
TxEd Reviewed by Jan and Cliff Kent
Amazing Directory A guide to the sources and resources
Amiga Developers A listing of Suppliers and Developers
Public Domain Catalog A listing of Amicus and Fred Fish PDS
Dos 2 Dos review R. Knepper Transfer files from PC/MS-DOS
MaxiPlan review by Richard Knepper The Amiga Spreadsheet
Gizmoz by reviewed by Peter Wayner Amiga extras!
The Loan Information Program by Brian Catley
basic prog. to for your financial options
Starting Your Own Amiga Related Business by W. Simpson
Keep Track of Your Business Usage for Taxes by J. Kummer
The Absoft Amiga Fortran Compiler reviewed by R A. Reale
Using Fonts from AmigaBasic, Part Two by Tim Jones
68000 Macros on the Amiga by G. Hull Advance your ability.
TDI Modla-2 Amiga Compiler review by S Faiwisze

Volume 2 Number 1 1987

What Digi-View Is... Or, What Genlock Should Be! by J. Foust
AmigaBasic Default Colors by Bryan Catley
AmigaBasic Titles by Bryan Catley

AC Provides Amiga Support!

A Public Domain Modula-2 System reviewed by Warren Block
One Drive Compile by Douglas Lovell Lattice C with one drive
A Megabyte Without Megabucks by Chris Irving
An Internal Megabyte upgrade
Digi-View reviewed by Ed Jakober
Defender of the Crown reviewed by Keith Conforti
Leader Board reviewed by Chuck Raudonis
Roundhill Computer System's PANEL reviewed by Ray Lance
Digi-Paint.....by New Tek previewed by John Foust
Deluxe Paint II ....from Electronic Arts previewed by J. Foust

Volume 2 Number 2 1987

The Modem by Josph L. Rothman efforts of a BBS Sysop
MacroModem reviewed by Stephen R. Pietrowicz
GEMINI or "It takes two to Tango" by Jim Meadows
Gaming between machines
BBS-PC! reviewed by Stephen R. Pietrowicz
The Trouble with Xmodem by Joseph L. Rothman
The ACO Project....Graphic Teleconferencing on the Amiga by S. R. Pietrowicz
Flight Simulator II....A Cros Country Tutorial by John Rafferty
A Disk Librarian in AmigaBASIC by John Kennan
Creating and Using Amiga Workbench Icons by C. Hansel
AmigaDOS version 1.2 by Clifford Kent
The Amazing MIDI Interface build your own by Richard Rae
AmigaDOS Operating System Calls and
Disk File Management by D. Haynie
Working with the Workbench by Louis A. Mamakos Prog in C

Volume 2 Number 3

The Amiga 2000™ by J Foust
A First look at the new, high end Amiga™
The Amiga 500™ by John Foust
A look at the new, low priced Amiga
An Analysis of the New Amiga PCs by J. Foust
Speculation on the New Amigas
Gemini Part II by Jim Meadows
The concluding article on two-player games
Subscripts and Superscripts in AmigaBASIC by Ivan C. Smith
The Winter Consumer Electronics Show by John Foust
AmigaTrix by W. Block Amiga™ shortcuts
Intuition Gadgets by Harriet Maybeck Tolly
A journey through gadget-land, using C
Shanghai reviewed by Keith M. Conforti
Chessmaster 2000 & Chessmate reviewedby Edwin V. Apel, Jr.
Zing! from Meridian Software reviewed by Ed Bercovitz
Forth! by Jon Bryan Get stereo sound into your Forth programs.
Assembly Language on the Amiga™ by Chris Martin
Roomers by theBandito Genlocks are finally shipping, & MORE!!!
AmigaNotes by R. Rae Hum Busters... "No stereo? Y not?...
The AMICUS Network by J. Foust
CES, user group issues and Amiga Expo"

Volume 2 Number 4 1987

Amazing Interviews Jim Sachs by S. Hull Amiga Artist
The Mouse That Got Restored by Jerry Hull and Bob Rhode
Sluething Public Domain Disks with CLI by John Foust
Highlights: the San Francisco Commodore Show by S Hull
Speaker Sessions: San Francisco Commodore Show H Tolly
Household Inventory System in AmigaBASIC™ by B Catley
Secrets of Screen Dumps by Natkun Okun
Using Function Keys with MicroEmacs by Greg Douglas
Amigatrix II by Warren Block More Amiga shortcuts
Basic Gadgets by Brian Catley Create gadget functions
Gridiron reviewed by K. Conforti Real football for the Amiga
Star Fleet I Version 2.1 reviewed by J. Tracy Amigain Space
The TIC reviewed by J. Foust Battery powered Clock Calendar
Metascope review by H. Tolly An easy-to-use debugger

Volume 2 Number 5 1987

The Perfect Sound Digitizer review by R. Battle
The Future Sound Digitizer by W. Block Applied Vision's SD
Forth! by J. Bryancomparing JForth amd Multi-Forth.
Basic Input by B. Catley AmigaBASIC input routine for use in all your programs.

Volume 2 Number 5 1987 continued

Writing a SoundScape Module in C by T. Fay Programming with MIDI, Amiga and SoundScape by SoundScape author.
Programming in 68000 Assembly Language by C. Martin
Continuing with Counters & Addressing Modes.
Using FutureSound with AmigaBASIC by J. Meadows AmigaBASIC Programming utility with real, digitized STEREO
AmigaNotes Rich Rae reviews SoundScape Sound Sampler.
More AmigaNotes by R. Rae A furtherlook at Perfect Sound.
Waveform Workshop in AmigaBASIC by J. Shields edit & save
waveform for use in other AmigaBASIC programs.
The Mimetics Pro MIDI Studio by Sullivan, Jeffery
A review of Mimetics' music editor/player.
Intuition Gadgets Part II by H. MaybeckTolly Boolean gadgets
provide the user with an on/off user interface.

Volume 2 Number 6 1987

Forth! by J. Bryan Access resources in the ROM Kernal.
The Amazing Computing Hard Disk Review by J. Foust & S. Leemon In-depth looks at the C Ltd. Hard Drive, Microbotics' MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive and Xebec's 9720H Hard Drive. Also. a look at disk driver software currently under devlopment.
Modula-2 AmigaDOS™ Utilities by S. Faiwiszewsk A
Calls to AmigaDOS and the ROM kernal.
Amiga Expansion Peripheral by J. Foust
Explanation of Amiga expansion peripherals.
Amiga Technical Support by J. Foust
How and where to get Amiga tech support.
Goodbye Los Gatos by J. Foust Closing Los Gatos.
The Amicus Network by J. Foust West Coast Computer Faire.
Metacomco Shell and Toolkit by J. Foust A review
The Magic Sac by J. Foust Run Mac programs on your Amiga.
What You Should Know Before Choosing an Amiga 1000
Expansion Device by S. Grant
7 Assemblers for the Amiga by G. Hull Choose your assembler
Shakeup Replaces Top Management at Commodore by S. Hull
Peter J. Baczor" by S. Hull Manager at CBM gives an inside look

Logistix A review by Richard Knepper
Organize! by A review Richard Knepper database.
68000 Assembly Language Programming on the Amiga
by Chris Martin
Superbase Personal Relational Database by Ray McCabe
AmigaNotes by Rae, Richard A look at FutureSound
Commodore Shows the Amiga 2000 and 500 at the Boston
Computer Society by H Maybeck Tolly

Volume 2, Number 7 1987

New Breed of Video Products by John Foust...
Very Vivid! by Tim Grantham...
Video and Your Amiga by Oran Sands III
Amigas & Weather Forecasting by Brenden Larson
A-Squared and the Live! Video Digitizer by John Foust.
Aegis Animator Scripts and Cel Animation by John Foust
Quality Video from a Quality Computer by Oran Sands III.
Is IFF Really a Standard? by John Foust..
Amazing Stories and the Amiga™ by John Foust.
All about Printer Drivers by Richard Bielak
Intuition Gadgets by Harriet Maybeck Tolley.
Deluxe Video 1.2 by Bob Eller
Pro Video CG1 by Oran Sands III.
Digi-View 2.0 Digitizer/Software by Jennifer M. Janik
Prism HAM Editor from Impulse by Jennifer M. Janik

Volume 2, Number 7 1987 continued

Easyl drawing tablet by John Foust..
CSA's Turbo-Amiga Tower by Alfred Aburto
68000 Assembly Language by Chris Martin.

Volume 2, Number 8 1987

This month Amazing Computing™ focuses on entertainment packages for the Amiga. Amazing game reviews...

SDI, Earl Weaver Baseball, Portal, The Surgeon, Little Computer People, Sinbad, StarGlider, King's Quest I,II and III, Faery Tale Adventure, Ultima III, Facets of Adventure, Video Vegas and Bard's Tale.

Plus Amazing monthly columns... Amiga Notes, Roomers, Modula-2, 68000 Assembly Language and The Amicus Network.
Disk-2-Disk by Matthew Leeds
The ColorFonts Standard by John Foust
Skinny C Programs by Robert Riemersma, Jr.
Hidden Messages In Your Amiga™ by John Foust
The Consumer Electronics Show and Comdex.by J Foust

Volume 2 Number 9 1987

Analyze 2.0 reviewed by Kim Schaffer
Impact Business Graphics review by Chuck Raudonis
Microfiche Filer review by Harv Laser
Pagesetter review by Rick Wirch
Gizmoz Productivity Set 2.0 review by Bob Eller
Kickwork review by Harv Laser
Diga Telecommunications Package review by Steve Hull
Mouse Time and Timesaver review by John Foust
Insider Memory Expansion review by James O'Keane
Microbotics Starboard-2 review by S. Faiwiszewski
Leather Goddesss of Phobos by Harriet Maybeck-Tolly
Lattice C Compiler Version 3.10 reviewed by Gary Sarff
Manx 3.4a Update reviewed by John Foust
AC-BASIC reviewed by Sheldon Leemon
AC-BASIC Compiller an alternative comparison by B Catley
Modula-2 Programming S Faiwiszewski Raw Console Dev. Events
Directory Listings Under AmigaDOS by Dave Haynie
AmigaBASIC Patterns by Brian Catley
Programming with Soundscape Todor Fay manipulate's samples
Bill Volk, Vice-President Aegis Development, by Steve Hull
Jim Goodnow, Devekoper of Manx 'C' interview by Harriet M Tolly
Plus a great collection of monthly columns...

Volume 2 Number 10 1987

Max Headroom and the Amiga by John Foust
Taking the Perfect Screen Shot by Keith Conforti
Amiga Artist: Brian Williams by John Foust
Amiga Forum on Compuserve™... Software Publishing
Conference Transcript by Richard Rae
All About Online Conferencing by Richard Rae
dBMAN reviewed by Clifford Kent
Amiga Pascal reviewed by Michael McNeil
AC-BASIC Compiler reviewed by Bryan Catley
68000 Assembly Language by Chris Martin
Amiga Programming:
Amiga BASIC Structures by Steve Michel
Quick and Dirty Bobs by Michael Swinger
Directory Listings Under Amiga-DOS, Part II by Dave Haynie
Fast File I/O with Modula-2 by Steve Faiwiszewski
Window I/O by Read Predmore
Plus a great collection of monthly columns...

Volume 2 Number 11 1987

Word Processors Rundown by Geoff Gamble
ProWrite, Scribble!, and WordPerfect compared
LPD Writer Review by Marion Deland
VizaWrite Review by Harv Laser
Aedit Review by Warren Block
WordPerfect Preview by Harv Laser
Jez San Interview by Ed Bercovitz—StarGlider author speaks!
Do-it-yourself Improvements to the Amiga Genlock
Digi-Paint Review by Harv Laser
Sculpt 3D Review by Steve Pietrowicz
Shadowgate Review by Linda Kaplan
TeleGames Review by Michael T. Cabral
Reason Preview: an intense grammar examination application
As I See It by Eddie Churchill WordPerfect,Gizmoz V2.0 & Zing!
AmigaNotes by R Rae 4 electronic music books
Modula-2 Programming by S.Faiwiszewski devices, I/O, serial port
68000 Assembly Language by Chris Martin Display routines
The AMICUS Network by John Foust—Desktop Publishing, Seybold
C Animation Part II by Mike Swinger Animation Objects

BASIC Text by Brian Catley Pixel perfect text positioning
Soundscape Part III by Todor Fay VU Meter and more
Fun with Amiga Numbers by Alan Barnett
File Browser by Bryan Catley—Full Feature BASIC File Browsing
Plus a great collection of monthly columns...

Volume 2 Number 12 1987

The Ultimate Video Accessory by Larry White
The Sony Connection by Stewart Cobb
15-Puzzle in AmigaBASIC by Zoltan Szepsi
Life, Part I: The Beginning by Gerald Hull
The ultra-complex nine blit solution to the "Game of Life."
Amiga Virus! by John Foust
CLI Arguments in C by Paul Castonguay
MIDI Interface Adapter by Barry Massoni
Amiga 1000-style MIDI interfaces can fit A2000s or 500s
Modula-2 by S. Faiwiszewski Part 1: command line calculator
AmigaNotes by Rick Rae audio changes made in the A500 &A2000.
Animation for C Rookies: Part III by M. Swinger double-buffering.
The Big Picture by Warren Ring Assembly language programming
Karate Kid Reviewby Stephen R. Pietriowicz
GO! 64 review by John Foust, James O'Keane, and Rick Wirch
Three C-64 experts investigate a new Amiga 64 emulator.
A-Talk-Plus Review by Brendan Larson
Calligrapher Review by John Foust
Animator: Apprentice Review by John Foust
Playing Dynamic Drums on the Amiga by David N. Blank
WordPerfect Review by Steve Hul
Insider/Kwikstart Review by Ernest P. Viveiros Sr
RAM & ROM expansion: Comments and installation tips.
Forth! by Jon Bryan DumpRPort utility for your Multi-Forth toolbox.
As I See It by Eddie Churchill Digi-Paint, Portal, &Videoscape 3D.
The Commodore Show and AmiExpo: New York!
Plus a great collection of monthly columns...

Volume 3 Number 1 1988

AmigaNotes by Richard Rae Amiga digital music generation .
C Animation Part IV by Michael Swinger
Forth by John Bryan Sorting out Amiga CHIP and FAST memory
The Big Picture by Warren Ring Daring assembler language programming; CLI system calls and manipulating disk files.
68000 ASssembly Langueage Programming by Chris Martin
"Create a multi-color screen without using Intuition routines!"
Modula-2 Programming by S. Faiwiszewski A new modula-2!
Amicus Network Special Report: Fall COMDEX by J. Foust
The ultimate Video Accessory: Part II by Larry White
Life: Part II by Gerald Hull The Amiga blitter."
FormatMaster: Professional Disk Formatting Engine byC.Mann
Put Batch language to work on the drudgery of disk formatting .
BSpread by Brian Catley full featured AmigaBASIC spreadsheet!
AmigaForum Transcript ed. by Rick Rae Amiga's Dave Haynie.
Haicalc Review by Chuck Raudonis easy to use, spreadsheet.
VIP Professional Review by S. Mitchell Manage stock porfolio
Money Mentor Review by S.Kemp Personal finance system.
Investor's Advantage Review by Richard Knepper
plus "Poor Man's Guide to the Stock Market."
Plus a great collection of monthly columns...

Volume 3 Number 2 1988

Laser Light Shows with the Amiga by Patrick Murphy
Lasers and the Amiga: A Dazzling Tandem
The Ultimate Video Accessory: PartIII by Larry White
Take the final stepstoward designing your own videos.
Our First Desktop Video by Larry White
Step-by -step guide to organizing & presenting your Amiga video.
Hooked on the Amiga with Fred Fish interview by Ed Berkovitz.
Photo Quality Reproduction with the Amiga and Digi-View
by Stephen Lebans
Balancing your Checkbook with WordPerfect Macros by S.Hull
Hand your checkbook worries over to the Amiga.
More Basic Text by Bryan Catley easier text on an Amiga screen
Life: Part III by Gerald Hull
Sries winds up with famed nine-blit calculation & source to LIFER.
Solutions to Linear Algebra through Matrix Computations
by Ropbert Ellis
Simplify matrix algebra with basic operations & routines.
Modula-2 Programming by Steve Faiwiszewski
Catching up with Calc-a source follow-up.
68000 Assembler Language Programming by Chris Martin
Graphics- Part II of Assemgram.
Arazok's Tomb interview by Kenneth E. Schaefer
AIRT by S. Faiwiszewski innovative icon-basede program. lang.
Forms in Flight by S. Pietrowicz Render & Animate 3D objects
Silicon Dreams and the Jewel of Darkness by K E. Schaefer
Leisure suit Larry by Kenneth E. Schaefer
Two New Entries From Microbiotics by John Foust
M501 Expansion & Starboard II MultiFunctionboard.
Mindlight 7 and People Meter by John Foust
Phantasie Ken E. Schaefer Amazing Phantasie Character Editor.
Plus a great collection of monthly columns...

Volume 3 Number 3 1988

Desktop Video,Part IV by Larry White
Put all the pieces together-the desktop video commercial.
The Hidden Power of CLI Batch File Processing by J. Rothman
Make your Amiga easier to use with CLI Batch files.
A Conference with Eric Graham edited by John Foust
The mastermind behind Sculpt 3D and Animate 3D.
Perry Kivolowitz Interviewed by Ed Bercovitz Amiga insights from a major developer and personality.
Jean "Moebius" Giraud Interviewed by Edward L. Fadigan
Avant-garde art comes to the Amiga-in dazzling form.
PAL Help by Perry Kivolowitz A1000 expansion reliability.
Boolean Function Minimization by Steven M. Hart
A useful digital design tool in AmigaBASIC.
Amiga Serial Port and Midi Compatibility for Your A2000! by L. Ritter and G. Rentz Add an A1000-style serial port to the A2000!
Electric Network Solutions the Matrix Way by Robert Ellis
Engineers! Practicle routines for using matrix algebra.

The A.M.U.G. BBS List compiled by Joe Rothman, Chet Solace, & Dorothy Dean 514 BBS phone numbers in the U.S. & Canada.
FACC II reviewed by Graham Kinsey Speed your floppy drives.
Uninvited reviewed by K E. Schaefer
Flow reviewed by Pamela Rothman brainstorms into mental art.
Benchmark Modula-2 Compiler reviewed by Richie Bielak
Modula-2 Programming by Steve Faiwiszewski
The gameport device and simple sprites in action.
AmigaNotes by R. Rae A1000! Software-switchable output filter.
Roomers by The Bandito AmiExpo, Kickstart 1.4, Commodore
The Big Picture by Warren Ring-- Unified Field Theory!!
Plus a great collection of monthly columns...

Volume 3 Number 4 1988

Highlights from AmiExpo, Los Angeles by Steve Hull
Writing a SoundScape Patch Librarian T. Fay System Exclusive
Upgrade Your A1000 to A500/2000 Audio Power—by H.Bassen
Modifications to help your A1000 make sweet music, too!
Amiga Audio Guide Listing of all Amiga audio products.
Gels in Multi-Forth by John Bushakra
Macrobatics by Patrick J. Horgan Ease the trauma of assembly language programming.
Amiga Audio Sources The folks behind all those audio products.
Take Five! by Steve Hull five Amiga games reviewed.
Amiga Notes by Rick Rae A basic tour of Amiga audio.
The Ultimate Video Accesory, Part V by Larry White
Bug Bytes by John Steiner
The Big Picture by Warren Ring Part II Unified Field Theory.
Roomers by The Bandito
Hardware hijinx ... Toasted video ... the dream Amiga ... and more!
In the Public Domain by C.W. Flatte
Time Bandit review by Keith Conforti
AudioMaster review by B. Larson Real-time digitizing samples.
Music Mouse review by J Henry Lowengard
Making music without lifting a finger from the mouse.
Amiga-Tax/Canadian Version review by Ed Bercovitz
A Canadian income tax planning, preparation, & analysis package.
SAM BASIC review by Bryan Catley
A new BASIC which exploits even more unique Amiga features.

Volume 3 Number 5 1988

Interactive Startup Sequence by Udo Pernisz
The Command Line part 1by Rich Falconburg
AmigaTrix III by Warren Block—Tips and tidbits to ease Amiga life
Amiga Product Guide: Hardware Edition
Proletariat Programming by P Quaid—Public domain compilers
The Companion by P.Gosselin Amiga's Event Handling capability.
MindLight 7 reviewed by David N. Blank
VideoScape 3-D 2.0 reviewed by David Hopkins
Extend reviewed by Bryan D. Catley—An AmigaBASIC extension
AssemPro reviewed by Stephen Kemp
Opening a door to assembly language programming.
APL.68000 reviewed by Roger Nelson
Book Reviews by Richard Grace—Three "C" programming texts.
CBTREE reviewed by Michael Listman
A tidy collection of functions to aid the C programmer.
The Big Picture by Warren Ring
The three-part Unified Field Theory winds up
Modula-2 by Steve Faiwiszewski
Termination modules for Benchmark and TDI compilers.
68000 Assembly Language by Chris Martin
Peeling away the complication of display routines.
Plus a great collection of monthly columns...

Volume 3 Number 6 1988

Bear Time Reviewed by Steve Carter
What makes this inexpensive A1000 battery-backed clock tick?
Acquisition Reviewed by David N. Blank
A look inside the latest release of a powerful relational database.
Butcher 2.0 Reviewed by Gerald Hull
A tidy collection of diverse image processing utilities.
Reassigning Workbench Disks by John Kennan
Endless disk swapping comes to a merciful end.
Product Guide: Software Tools Edition
A listing of all the products you need to put your Amiga to work.
An IFF Reader in Multi-Forth by Warren Block
Create an easy to use IFF reader in Multi-Forth.
Basic Directory Service Program by Bryan Catley
A programming alternative to the GimmeeZeroZero windows.
C Notes from the C Group by Stephen Kemp
A beginner's guide to the power of C programming.
An Amiga Forum Conference with Jim Mackraz
The Amiga market as seen by the "Stepfather of Intuition."
Son of Seven Assemblers Reviewed by Gerald Hull
A ocmparative battle between seven native-code assemblers.
The 1988 Commodore Amiga Developers Conference
A look inside the conferences held in Washington, D.C.
Amiga Working Groups by Perry Kivolowitz and Eric Lavitsky
An outline of the innovative Amiga Working Groups concept.
The Command Line by Rich Falconburg
Exploring the multi-talented LIST command.
Plus a great collection of monthly columns...

Volume 3 Number 7 1988

Look, Up On the Screen, It's an Ami... It's a Pro... It's SuperGen
reviewed by Larry White—Genlock comparisons
An Interview with "Anim Man," Gary Bonham by B. Larson
An animated conversation with the man behind the format.
The Amiga at Spring COMDEX in Atlanta by Ed Bercovitz
Amiga Product Guide: Video/Graphics Edition
Thirteen pages devoted to the Amiga's dazzling strong suit.
The Developing Amiga by Steve Pietrowicz
Developers' notes: PD vs. shareware vs. freely distributableetc.
Roll Those Presses! by Barney Schwartz
Welcome to the dandy, demanding world of desktop publishing!
Linked Lists in C by W. E. Gammill Put dynamic memory to work!
FrameGrabber Preview by Oran Sands
Capturing an image can now be as fast as punching a single key!

Amazing Volumes!

A First Look at Interchange reviewed by David Hopkins
Bridge the gap between those incompatible animation packages.
Perfect Vision reviewed by Bryan Catley
Capture, digitize and save pictures from any video source.
ProWrite 2.0 Review reviewed by Pamela Rothman
A graphic word processor specializing in efficient editing.
Doug's Math Aquarium: The Art of Mathematics by R. Bielak
Bear Products MegaRex II Expansion RAM by Steve Carte
The Command Line by Rich Falconburg
Amiga Notes by Rick Rae
The Other Guys' Synthia: A digital synthesizer features won't stop.
C Notes from the C Group by Stephen Kemp
Weathering the unknown "C" of basic object and data types.
Plus a great collection of monthly columns...

Volume 3 Number 8 1988

The Command Line by Rich Falconburg
The journey into the CLI continues.
The Developing Amiga by Stephen R. Pietrowicz
A gaggle of great programming tools.
Modula-2 Programming by Steve Faiwiszewski
Libraries and the FFP and IEE Math Routines.
C Notes from the C Group by Stephen Kemp
Arrays and pointers unmasked.
Dark Castle reviewed by Keith Conforti—The Black Knight lurks
Ports of Call reviewed by Julie Landry
Leatherneck reviewed by Michael Creeden-Rambo's not so tough!
Capone reviewed by Joyce and Robby Hicks- Light Guns blaze
Casino Fever reviewed by Michael T. Cabral—Vegas on Amiga
Ferrari reviewed by Jeffery Scott Hall—Start your engine
Arkanoid reviewed by Graham Kinsey—"blockbuster"
Ebonstar by Keith Conforti —black hole treking.
Deluxe Productionsreviewed by Harv Laser—Video wizardry
Game Pizazz by Jeffery Scott Hall— Register your questions here.
TrackMouse by Darryl Joyce
Convert a standard Atari trackball into a peppy Amiga TrackMouse.
Amiga Interface for Blind Users reviewed by Carl W. Mann
An ingenious interface that opens the Amiga to even more users!
Video in the Sunshine State reviewed by Stephen R. Pietrowicz
RGB Video Creations hosts a video unveiling!
Amiga Product Guide: Games Edition
Tumblin' Tots by David Ashley—assemply language program.
Plus a great collection of monthly columns...

Volume 3 Number 9 1988

The Kideo Tapes by John Dandurand
A Georgia elementary school puts desktop video to work.
Speeding Up Your System by Tony Preston floppy disk caching
Amiga Product Guide: Education Edition
Everything you need to send your Amiga to the head of the class.
Computer Aided Instruction by Paul Castonguay
Authoring system in AmigaBASIC.
Gels in Multi-Forth, Part II: Screenplay by John Bushakra
Make the IFF converter from Part I easy to use-gadgets, menus,etc.
AmiExpo Midwest '88 by Michael T. Cabral
After taking the coasts by storm, the Amiga wows Chicago
Intellitype by Harv Laser—Learning to type made easy...and *fun?*
Shakespeare by Barney Schwartz—Desktop publishing in full color.
XSpecs 3D by Steve Hull—A new dimension in Amiga graphics.
AmigaNotes by Richard Rae—How IFF sound samples are stored?
Take Five! by Steve Hull—Beat the back-to-school blues !
The Command Line by Rich Falconburg—continuing tour of CLI.
Hot on the Shelves by Michael T. Cabral & Michael Creeden
What do you get when you combine intense war strategy with a monochrome monitor and desktop presentation? Check it out.
Bug Bytes by John Steiner
C Notes from the C Group by Stephen Kemp
Operators, expressions, and statements in C uncovered.
Roomers by The Bandito Can Apple IIgs Plus keep Amiga away?

Volume 3 Number 10 1988

A First Look At Deluxe PhotoLab reviewed by David Duberman
A paint package, poster-maker, and image processing program
DiskMaster reviewed by Steve Hull —file management utility.
DSM: A MC680000 Disassembler reviewed by Gerald Hull
Looking for easily modifiable, assembler-ready code?
FBasic Language System reviewed by Patrick Quaid
BASIC compiler and development system.
Hot on the Shelves by Michael T. Cabral—Deviant dice, gripping gray scales, color cartography, mauling modems, and much more.
The Command Line by Rich Falconburg
NEWCLI: A painless way to create a new console window.
The Developing Amiga by S. Pietrowicz Usenet—24-Hour News
C Notes from the C Group by Stephen Kemp —loops
Roomers by The Bandito WP wars, ignominious interfaces, & more
PD Serendipity by C.W. Flatte—Fred Fish collectionpasses 150.
Comparison of MultiScan Monitors by Steven Bender
Five multiscan alternatives square off on the desktop.
Record Keeping for Free-lancers:
A Superbase Professional Tutorial by Marion Deland
Record keeping system for free-lance photographers and others.
On The Crafting of Programs by David J. Hankins—A look at optimization kicks off a series of articles on programming savvy.
Bob and Ray Meet Frankenstein by Robert D'Asto—Create, animate, and metamorphose graphics objects in AmigaBASIC.
Digital Signal Processing in AmigaBASIC by Robert Ellis
Perform your own digital experiments with Fast Fourier Transforms.
HAM & AmigaBASIC by Bryan Catley—Pack your AmigaBASIC programs with many of the Amiga's 4096 shades!
CAI—Computer Aided Instruction: Part II by Paul Castonguay
The Editor program wraps up our authoring system in AmigaBASIC.

Volume 3 Number 11 1988

Desktop Publishing with Professional Page by Barney Schwartz
tutorial in document creation, plus some jazzy enhancements.
Game Pizzazz by J. Hall gaming hints, tips, high-score secrets.
Structures in C by Paul Castonguay C programming in an nutshell.
On The Crafting of Programs by D. Hankins speed up your progs.
Desktop Video VI: Adding the Third Dimension by Larry White
Unraveling the complexity of 3D for your video creations.
A2000 Hard Drive Round Up by Sheldon Leemon
Keyclick by Mike M. Duppong a typewriter click in your keyboard.
More Linked Lists in C: Techniques and Applications by Forest W. Arnold Procedures for managing lists, storing diverse data types in the same list, and putting lists to work in your programs.
BASIC Linker by Brian Zupke Combine individual routines from your program library to create an executable program.
The Developing Amiga by Steven Pietrowicz
A look at mysteries and successes behind efficient beta testing.
Modeler 3D Preview reviewed by David Hopkins
A peek inside a new, open-ended 3D package.
AProDraw Graphics Tablet reviewed by Keith Conforti
Artists! Meet the future of Amiga graphics.
StarGlider II reviewed by Jeffery Scott Hall
Those irritating Ergons are back for another laser-lashing.
WShell reviewed by Lawrence Lichtman CLI substitute,
Hot on the Shelves by M. Cabral viruses, music, microfiche mastery
PD Serendipity by C.W. Flatte Fred Fish disks 149-152.
Roomers by The Bandito Golden RAM, 16-bit videogames, CD-I, another HAM skirmish ... what could possibly be NeXT?

Volume 3 Number 12 1988

Hot on the shelves by M. T. Cabral
Graphic adventure, control over Preferences, a Postscript print utility, sequence live action animation, a new deal for user groups and the figure construction set.
PD Serendipity by C. W. Flatte
Fred Fish disks 156-162
Bug Bytes by John Steiner
All the latest from the world of bugs and upgrades.
Roomersby The Bandito
AmiExpo, C.D. the latest from Commodore and more.
AmiExpo California By Stephen Kemp
Hot— All the news.
EMPIRE reviewed by Stephen Kemp
EMPIRE , the game of conquest, has finally come to the Amiga.
Virus Infection Protection (V.I.P.) reviewed by Jeffery Scott Hall
What makes a computer sick and the cure.
The Command Line by Rich Falconburg
What to do when the commands of AmigaDos fail.
Converting Patch Librarian Files by Phil Saunders
How to get your sounds from there to here.
E.C.T. SampleWare by Tim Mohansingh
The E.C.T. samples contain several gems.
The Creation of Don Bluth's Dragon's Lair by Randy Linden
A look behind the scenes.
Easy Menus in JForth by Phil Burk
HELLO WORLD.
Extending AmigaBasic by John Kennan
The use of library calls from within AmigaBASIC.
Better Dead Than Alien reviewed by Jeffery Scott Hall
Don't fire until you see the greens of their eyes.
Getting Started In Assembly by Jeff Glatt
An introduction to Amiga assembly language programming
AC/BASIC 1.3 reviewed by Bryan Catley
Release 1.3 of Absoft's AC/BASIC compiler for the Amiga.
Thexder reviewed by Bruce Jordan
Thexder turns out to be a real screamer: Action, Adventure, Fantastic Sound, and stunning Graphics.
Magellan: The AMIGA Gets Smart reviewed by Steve Gillmor
The worlds of artificial intelligence comes to the AMIGA in the form of A.I. system software.
C Notes From The C Group by Stephen Kemp
Program or function control coding; the case history.
AmigaDos, Assembly Language, And FileNotes by Dan Huth
Weapons in the war against file overload; accurate, descriptive file naming.

Volume 4 Number 1 1989

The Wonderful World of Hashnique reviewed by Shamms Mortier
A review of the Amiga software products of Hash Enterprises
Desktop Video by Richard Starr
Thinking about getting into Video? Here's what you'll need to know.
Industrial Strength Menus by Robert D'Asto
Add some snazzy submenus to your AmigaBASIC cuisine
Second Generation 2D Animation Software by Geoffrey Williams
Cel Animators and Key Frame Animators,
how they differ and a look into their use.
What's The Diff? reviewed by Gerald Hull
A review of Lattice's Compiler Companion
Scrolling Through SuperBitMap Windows by Read Predmore
How to implement SuperBitMaps
for viewing or drawing into large graphic areas.
Alive in 3D by Shamms Mortier
A review of Caligari, a High-End 3D sculpting & animation package.
Sync Tips by Oran J. Sands III
Dot crawl, the Amiga and composite video devices.
How May I Animate Thee?, Let Me Count The Ways-
by Shamms Mortier
An overview of animation techniques.
Stop-Motion Animation On The Amiga by Brian Zupke
A hands on approach to animation and the Amiga.
Roomers by The Bandito
Commodore's deal, RAM chip crisis, and more!
C Notes From the C Group by Stephen Kemp
Structures - A powerful feature of C
On the Crafting of Programs by David J. Hankins
What Format is right for you
The Command Line by Rich Falconburg
A look at new and improved Assembly Language commands
Questron II reviewed by Jeffery Scott Hall
Questron II - It's a journey back in time
Pointers, Function Pointers, and Pointer Declarations in C by Forest W. Arnold Reducing data type dependencies
Las Vegas Comdex Report by Louise Brinkmann
Commodore's new 2500, 2500 UX, and more!
Philadelphia World of Commodore by Chris Darsch & Rick Rae
Highlights of Philadelphia's Commodore Show
STELLARYX Review by Stephen Kemp
Exciting & challenging! Terrific stereo and sound effects
Arkanoid Imposters: Unmasking the Impostors. reviewed by Jeffery Scott Hall
A look at Arkanoid look-alikes
Bug Bytes by John Steiner
Bugs and upgrades
Death of a Process by Mark Cashman
Develop an error handling module in Modula-2

Volume 4 Number 2 1989

MovieSetter: The Next Generation by Steve Gillmor
A new level in Amiga animation.
AmiForum by Steve Pietrowicz
The Amiga pays a mid-winter visit to Mickey's hometown.
Max Morehead Interview by Richard Rae
Rick talks to the creator of Moviesetter.
A Common User Interface for the Amiga by Jim Bayless
Does the Amiga need a kinder, gentler interface?
Superbase Professional by Marion Deland
A user's look at Superbase Pro.
Microfiche Filer Plus by Ronald Courrier
A multi-tasking database that uses ARexx to work with other progrs
Torch 2081 by Jeffery Scott Hall
Fast action, and smooth graphics
SPY by Steve Faiwiszewski
Programming intrigue in Modula -2
Sync Tipsby Oren Sands
Getting inside the genlock.
On the Crafting of Programs by D J. Hankins
Do we need a common standard for C programming? Look ANSI.
C Notes from the C Group by Steven Kemp
An introduction to unions
The Command Line by Rich Falconburg
ED, your Workbench Screen Editor
An Introduction to ARexx programming by Steve Faiwizewski
Climbing the Towers of Hanoi.
Crunchy Frog by Jim Fiore
Amiga-specific C programming.
Super Bug Bytes by John Steiner
A double dose of bug repellant.
New Products by Michael Creeden
What's New? Deluxe Print II, The Talking Animator, 3 digit fun, more.
Roomers by The Bandito
Big resignation at Commodore, and the Bandito's fearless predictions.

Volume 4 Number 3 1989

Falcon reviewed by Joe DiCara
Latest from Spectrum Holobyte's hangar
Air Warrior reviewed by Michael Mantino
Hook up via modem and battle it out with international opponents!
Carrier Command reviwed by Lawrence Lichtmann
Admiral Lichtman signs up for carrier duty
World of Commodore Toronto by Ed Berkovitz
Ed files his report from Toronto
Fractal Fundamentals by Paul Castonguay
Experiment on the edge of a new science: create your own fractals
Image Processing with Photosynthesis by Gerald Hull
An experiment with a repertoire of AI image-processing techniques
Gizmoz 2.0 by Steve Carter
Steve Carter reviews version 2.0 of Digital Creations' box o' fun
AmigaTEX by Barney Schwartz
A page description language from Radical EyE software
flickerFixer by Steve Bender
Steve gives us a clear perspective on MicroWay's flickerFixer
Benchmark 1: Fully utilizing the MC68881 by Read Predmore
Part I: Turbocharging the savage benchmark
Breaking the Bmap Barrier by Robert D'Asto
Streamline your AmigaBASIC library access with Quick—Lib
Double Play by Robert D'Asto
AmigaBASIC program yeilds double vision.
C Notes From the C Group by Stephen Kemp
A walk through preprocesor control lines
New Products...and other Neat Stuff by Michael Creeden
3-M-dous! The latest from MichTron, MicroEd and Mindscape
The Video Desk by Larry White
The Amiga meets Nikon Camera
Roomers by The Bandito Magneto-optical disks on the Horizon?...
Amiga first stop for WordPerfect 6.0?...and the HAM paint wars.

Volume 4 Number 4 1989

AmiEXPO NY '89 by Steve Gillmor
Amiga takes a bite out of the Big Apple.
AmiEXPO Art and Video Contest Winners by Steve Jacobs
Winners of the first-ever AmiEXPO art event.
Adding the Not-So-Hard Disk by J P. Twardy
Short on memory? Installing a Hard Drive was never so easy.
Hard Drives—an Introduction by Jon A. Boulle
A straightforward, "no-Boulle" comparison of Hard Drives.
Tax Break by Kim Schaffer
OXXI's new tax program.
The Max Hard Drive Kit by Donald W. Morgan
A Hard Drive installation project, using Palomax's Max kit.
Menace reviewed by Jeffrey Scott Hall
Blast aliens & destroy the evil Draconia.
Sync Tips by Oran J. Sands
Oran presents a clearer picture of video and computer resolutions.
Passing Arguments by Brian Zupke AmigaBASIC subprogram explains step-by-step how to pass data from the CLI to AmigaBASIC
C Notes from the C Group by Stephen Kemp
A humorous view of the wacky world of programmers.
Creating a Shared Library by John Baez
Program for increased Amiga productivity.
MultiSort by Steve Faiwiszewski
Put your data in its place with this ultra organizing program.
New Products and Other Neat Stuff by Michael Creeden
Face-off with Gretsky, play Picasso with Basic Art Encoder, balance your Desktop Budget, plus more.
Snapshot by R. Brad Andrews
Four exciting Amiga games are reviewed .
Roomers by The Bandito Amiga developers go for the bucks, Commodore stock up, & Paint Wars.
Bug Bytes by John Steiner
Soft-Logik's PageStream, and more.

Volume 4 Number 5 1989

The Business of Video by Steve Gillmor
Get started in the video business.
An Amiga Adventure by Larry White
The globetrotting Amiga in Cologne, Germany.
Uninterruptible Power Supply (UPS), Part I by Steve Bender
Voltage spikes, surges, power failures? Are they uncommon?
The Amazing Audio Digitizer by Andre Theberge
Quality Amiga audio for less—building your own stereo digitizer.
A MIDI Out interface by Br. Seraphim Winslow
Helpful tips for happy jamming.
Digitized Sounds in Modula-2 by Len A. White
Produce impressive sound effects with sampled sounds.
Sync Tips by Oran J. Sands
The secrets hidden beneath the flicker mode.
On the Crafting of Programs by David J. Hankins
See how Lattice C 5.02 measures up.
Insta Sound in AmigaBASIC by Greg Stringfellow
The sounds you want for your program—in an instant!
Who are you, Mr. Guru? by David Martin
David exposes this Amiga deviant for what he really is.
Gold Disk's Professional Draw by R. Shamms Mortier
The latest in professional drawing tools from Gold Disk.
Electronic Arts' DeluxePaint III by David Duberman
DPaint's paintbrush grows feet—combines paint with animation.
Aegis's AudioMaster II by Phil Saunders
Aegis's newest rendition of sound sampling & editing is reviewed.
New Wave Software's Dynamic Studio by Chuck Raudonis
New Wave's on a roll with this follow-up to Dynamic Drums.
Dr. T's MIDI Recording Studio by Tim Mohansingh
A high-performance, low-budget remedy for your MIDI ills.
Snapshot by R. Brad Andrews
Alien Syndrome and Tetris are among the new Amiga games.
New Products and Other Neat Stuff by Michael Creeden
Central Coast Software calls huddle over phony Qback 3.0, daVinci meets Disney with DPaint III, Blue Ribbon Bakery serves up organization, plus more.
Bug Bytes by John Steiner
VirusX 3.3 an evil twin, some bickering from Nag Plus 3.0, plus.
Roomers by the Bandito The Bandito stalks AmiEXPO NY, Atari/Nintendo lawsuit expands, & the Beatles get a little help from the law
PD Serendipity by C.W. Flatte
C.W. covers Fred Fish from 189-200.
C Notes from the C Group by Stephen Kemp
Formatted output functions.

Volume 4 Number 6 1989

New Products & Other Neat Stuff by Michael Creeden
Add another dimension to your Amiga with Design 3D, Escape from the abandoned planet Atrax, and more!
Adventures in ARexx by Steve Gillmor
Enter the world of multitasking with a powerful super-application.
NAG Review by Marion Deland
An electronic appointment calendar with a sense of humor.
Digi-View Gold - It's Gold!by Bruce Jordan
A review of NewTek's video digitizing system.

Bug Bytes by John Steiner A look at some problems with the A2090 controller card and high-resolution & more.
KindWords 2.0 review by Marion Deland
High-quality fonts plus graphics, at the expense of speed.
PageStream tutorial Part I by Barny Schwartz
A look at Soft-Logik's full-featured document processor.
Video Preperation by Otto Focus
How to plan your video before you go on vacation
Roomers by The Bandito The Bandito takes a look at Commodore's future and the end of the Apple II.
At Your Request by John F. Weiderhim
Design your own requesters in AmigaBASIC.
Exploring Amiga Disk Structures by David Martin
A look at the heart of the Amiga: AmigaDOS. 58
CMI Accelerator Processor review by Rich J. Grace
Boost the performance of your Amiga at a low cost. 63
Diskless Compile in C by Chuck Raudonis Make development easy with COMPILE, a full-featured programmer's workbench.
(UPS), Part II by Steve Bender Steve continues his discussion on the technical aspects and details of various types of UPS units.
Programming the '881 Part II by Read Predmore
A discussion on how to calculate Mandelbrot & Julia sets.
C Notes from the C Group by Stephen Kemp
Steve discusses some ways to avoid problems when passing parameters between functions.
PD Serendipity by C.W. Flatte
C.W. covers Fred Fish disks 201-210.

To be continued........

video, Schmideo

by Barry Solomon

It all started with my friend, Simon. We met while working at a fancy audio/video store. We got along right off the bat. I loved listening to him talk about his computer. At the time he had an Amiga 1000 and was about to trade it in for a 2000. I guess I wasn't the only one he mesmerized with his tales; two other guys in his department had recently purchased Amiga 500's. They loved to talk about their Amigas, but when Simon spoke about his... Well, there was a certain gleam in his eyes. He was like a sixteen-year-old boy who had just convinced that special girl to accompany him to the prom. Whenever we had a moment to spare at work, he would tantalize me with stories about his amazing machine.

He talked about computer graphics, animation, and ray-tracing—all of which I knew little or nothing about. Each time we spoke, I became more and more curious about this mystical device. He brought in a few issues of *Amazing Computing.* I became more fascinated each day and was soon borrowing copies to read at home. I began to question him daily about articles I had read. He could not supply me with enough magazines. In a few weeks he was bringing them to work in bundles, and my questioning, I'm afraid, turned into grilling. Always the good sport, Simon took it in stride.

Then one magic day Simon brought that infamous Amiga demo tape into work. I must have watched it at least 20 times on our best monitor with my mouth hanging open. I had always been interested in graphics. In my younger days, I had even gone so far as to make a few attempts at film animation. I suppose I never followed it through, because I never really considered myself an artist.

By now I could hear the wheels turning inside my head. I could almost smell the smoke I imagined must have been pouring out of my ears. Hadn't I always been fascinated by colors and shapes and motion? Hadn't I always wanted to be creative? Didn't I have a brother-in-law who ran the video department at a local hospital, and did video work of his own on the side? And couldn't he, just perhaps, use some computer graphics???

I spoke with my brother-in-law after work that day, bringing the subject up oh-so-carefully. It turned out that, yes, he thought it was a great idea and he could probably budget some money into his projects for computer graphics! Well, that was all I needed to hear! I was off and running. I began to purchase my own copies of *Amazing Computing.* In fact, I bought every Amiga magazine I could find. Not wanting to make any mistakes, I even bought magazines on PC's and Macs. My questioning-turned-grilling of Simon became a regular inquisition, but he took it like a champ.

I was about four months into my quest when I received my second omen. I was roused from my lunchtime reverie one day to help answer an unusual question from a customer. It seems someone wanted to know if our VCRs worked on NTSC video. Now to be fair to the other salespeople, they all knew that our VCRs were NTSC. I guess they were just thrown by the question, because all VCRs in this country are NTSC. In a sense, it was kind of like asking, "Does this hair dryer plug into the wall?" or "Is this peanut butter made from peanuts?"

I ran upstairs and introduced myself to a very pleasant-looking fellow. I explained that all our VCRs were NTSC and that, in fact, all American television of any type or format was NTSC. The customer quickly informed me that what he really wanted to know was whether the VCRs had an NTSC video input jack.

My curiosity was roused. "If you don't mind my being nosey, why do you ask?"

My heart began to race as he explained that he had a computer set-up with an NTSC output and he wanted a good VCR to record some graphics.

I think my voice must have cracked as I asked, "What kind of computer?"

"An Amiga," he answered.

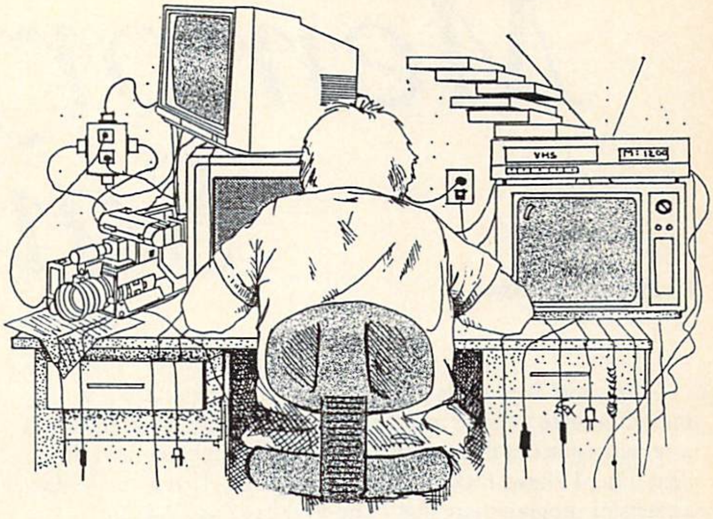
"Oh, which one?"

He laughed and said, "Well, I have a couple of 500's, a couple of 1000's, and a few 2000's."

I must have been staring. "Just who the hell are you?" I stammered.

"I'm Don Hicks, Managing Editor of *Amazing Computing.* Pleased to meet you."

If I needed any further convincing, that was it! The gods were definitely smiling down upon me. Like Steve



OR, THE LAYMAN'S GUIDE TO AMIGA VIDEO

Martin in *The Jerk*, I had a "Special Purpose." I began in earnest the Quest for Funds. Friends, neighbors, relatives—no one was safe. I begged, borrowed, weaseled, and cajoled my way into a modest 2000 system.

The first thing that occurred to me was that, not being independently wealthy, this would have to be a sideline for me, at least for a while. I realized that most of the video professionals who had authored the many articles and columns I read were indeed professionals and had been in the video business for quite some time. The Amiga may have been new to them, but video was not. Basically, I was starting from scratch, although I did have the distinct advantage of having a connection (my brother-in-law) to whom I could pitch my services.

Almost a year passed between the day my interest began and the day I picked up my Amiga. Until that day I had never actually seen an Amiga. I had researched it up, down, and sideways but had never really touched one. I would venture to say that of all the people on this planet who have never used an Amiga, I knew more about it than anyone.

I have been in business as Viewtopia Computer Graphics for about nine months now and have completed six projects. (I am now working on numbers seven and eight.) What I have learned—mostly by trial and error (with emphasis on the error)—could fill a book. What I still have to learn could fill volumes. What I have earned... Well, I didn't get into this for the money anyhow (but I still have hope!).

I have written all this for a couple of reasons. First, it seems that too many of the articles I read have been written by people burned out on the Amiga. They are disappointed in this one's lack of support, or that one's lack of support; they are upset with this program or that company, etc., etc., *ad nauseum*.That's not me. No sir, I love the Amiga. The honeymoon is far from over for me and my dear electronic friend.

Second, the articles I read always seem to be written by two types of people—those who don't know video from a hole in the ground, or the super-professional video types who only use the Amiga while their Crays are in the shop!

In the last two weeks alone I have met two people who recently purchased Amigas with the idea of producing some sort of video graphics. Both of them are rank amateurs, fresh out of the box. Strangely, both are middle-aged (the onset of a second childhood, perhaps?). They had both read many Amiga magazines and many articles on the Amiga, but neither one had a clue! It seems half these articles told them nothing of any value, and the other half just confused them. Neither of these men appear to be lacking in intelligence. It's just that no one is writing truly informative video articles for video beginners.

If you want to know about sub-carrier phase, vertical switching, and horizontal sync, great! But you don't need to know these things to be creative, or even to make a little money. What you do need to know is, "How do I do *this*?" or "Does *this* work?—and if it does, does it just work okay, or does it work exceedingly well?"

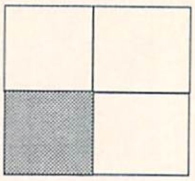
These days, when every revision of a program becomes a "dazzling new product" and every genlock from $150 up produces "broadcast-quality output," beginners can become confused. I know I was (and still am!). What I will try to do in future articles is give you the beginner's-eye view of professional video. Or the professional approach to beginner's video. Or... well, you get the idea.

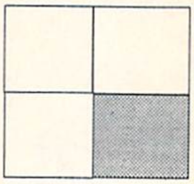
I will do my best to give you simple, useful hints for the types of problems you are likely to come across. Please write if you have questions about video or video-related graphics that require practical solutions (not charts, graphs, and formulae). I will do my best to answer your questions in simple, practical, down-to-earth terms so you can continue to enjoy your Amigas as the gods intended.

•AC•

Barry Solomon, Video Editor
c/o Amazing Computing
P.O. Box 869
Fall River, MA 02722-0869

Memory Squares





by Mike Morrison

Memory Squares is a game I wrote in AmigaBASIC. It started out like most of my programs. I was playing around with a certain aspect of the Amiga, like sound or graphics, and the next thing I knew, I had written a new game. That is one of the aspects of programming that I find very exciting. You are limited only by your imagination (and the number of hours in the day).

Memory Squares is a complete program in the sense that it is functional. The number of enhancements that could and should be added are limitless. Loosen up your tie and let your mind wander! Modify, change, add, delete, cut, paste, append, criticize, modernize, and expand this code to your hearts content. Get into the nuts and bolts of the code and make it a real learning experience. How many times do you think the Amiga beeped at me with syntax errors while I was writing this?

Gameplay

The game play is basic, and I have found it to be somewhat addictive. The game board is made up of a large square that is divided into four smaller squares. The game starts out by lighting two squares randomly and then waits for you to repeat the same sequence. As the game progresses more and more squares are lit, and you need to remember the sequence and repeat it.

To add interest to the game play, the sequences pick up speed as the game progresses. If you complete the sequence correctly you are rewarded with a small 'victory' display. You have four 'lives' to use before the game ends. If you have a good memory, you could be added to the high score list.

The Source

The nice thing about programming a game is that you can experiment and learn while you go. You also end up with something fun that other people can enjoy and learn from.

Coding the program

Set the seed for the RND() function so it will return a different sequence of numbers when the program is started again.

```
RANDOMIZE TIMER
```

Reserve space so we can keep track of square sequence the program chooses, including the high score list.

```
DIM where(10),scores(11),names$(11)
```

The ON..ERROR allows us to capture any errors that may occur when the program is running while trying to fix whatever may be wrong. It is used in case the high score file doesn't exist. When the 'file not found error' occurs, the error is handled by skipping to the label 'skipit'.

```
ON ERROR GOTO skipit
OPEN "memory.score" FOR INPUT AS 1
x=0
WHILE NOT EOF(1)
  x=x+1
  INPUT #1,names$(x),scores(x)
WEND
CLOSE #1
```

If everything went OK, set the error handling back to the system, or else the next time any other error occurs, it will jump to the 'skipit' label. Then go directly to the 'start' label and get things moving.

```
ON ERROR GOTO 0
GOTO start
```

We end up here if there is no high score file, so fill it with 'Empty for now'.

```
skipit:
FOR x=1 TO 10
  'names$(x)="Empty for now"
NEXT
```

We first go to the 'endyet' subroutine because it waits for a response from the player before moving on. No need to write code that already exists in our program.

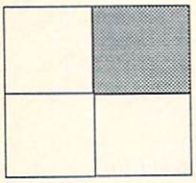
```
start:
GOTO endyet
```

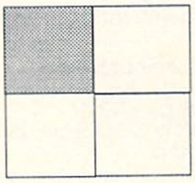
This is where it all happens. The 'x' variable keeps track of the number of rounds that will be played. Currently it is impossible to get through 10 rounds, the program moves to quick by then. You could slow things down by messing with the variable 'd' which controls the delay along with the 'pause' sub program described below.

```
main:
GOSUB init
FOR x=1 TO 10
  GOSUB board
  d=d-.1
  round=round+2
```

The 'z' loop is where the computer randomly picks the next square. We store it in array 'where()' so later we can check to see if the player picks the correct square.

```
FOR z=1 TO round
  r=INT(4*RND(1))+1           'Pick a number from 1-4.
  where(z)=r
  ON r GOSUB bx1,bx2,bx3,bx4  'Draw the box picked.
NEXT z
```





```
 GOSUB turn
 IF didit=0 THEN GOSUB minicelebrate
 IF noleft=-1 THEN endyet
NEXT x
```

If you make it through ten sequences, the program would end here. Currently it goes to the 'endyet' subroutine. If you modify the code the program can be completed ten times, and you may want to add a grand victory celebration here.

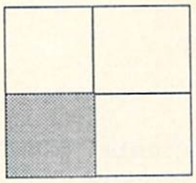
```
GOTO endyet
```

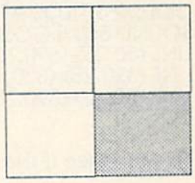
Set up a few odds and ends here. The three pieces that represent your number of turns left are drawn here.

```
init:
CLS
noleft=3:sc=0
FOR ml=1 TO noleft    'Draw number of turns left.
 LINE (10+(ml*60),160)-(60+(ml*60),180),2,b
 LINE (35+(ml*60),160)-(35+(ml*60),180),2
 LINE (10+(ml*60),170)-(60+(ml*60),170),2
NEXT ml
d=1.2:c=3:round=0
RETURN
```

This sub routine plays a tune and then asks if you want to play again. If your score is good enough to make the high score list, then the 'hiscore' routine is called. If you want to play again the program starts over. If you want to quit, the high score list is displayed and the program ends.

```
endyet:
FOR s=300 TO 500 STEP 20
 SOUND s,1
 SOUND 500-s,1
NEXT

IF sc>scores(10) THEN GOSUB hiscore

GOSUB board

LOCATE 3,25:PRINT"Select with mouse to play again."
COLOR 3,1:LOCATE 7,25:PRINT"YES"
COLOR 2,1:LOCATE 7,55:PRINT"NO":COLOR 1,0

GOSUB interpit
LOCATE 2,25:PRINT"                              "

IF pt=1 OR pt=3 THEN GOTO main
CLS
COLOR 2,1:LOCATE 2,28:PRINT "Memory Square Aces":COLOR
1,0
LINE (128,32)-(464,144),2,bf
COLOR 1,2
FOR x=1 TO 10
 LOCATE 6+x,20:PRINT names$(x)
 LOCATE 6+x,50:PRINT scores(x)
NEXT x
COLOR 1,0:LOCATE 1,1
END
```





If you made the list the appropriate space on the list is cleared, and you are asked to enter your name. The new list is then saved.

```
hiscore:
CLS:here=0
COLOR 2,1:LOCATE 2,28:PRINT "Memory Square Aces":COLOR
1,0
```

Find out where in the list your score belongs. Then blank the old name out.

```
FOR x=10 TO 1 STEP -1
 IF sc>scores(x) THEN
  scores(x+1)=scores(x):names$(x+1)=names$(x)
  scores(x)=sc:names$(x)="":here=x
 END IF
NEXT x
```

Display the list with the blank spot for the new name.

```
LINE (128,32)-(464,144),2,bf
COLOR 1,2
FOR x=1 TO 10
 IF x=here THEN COLOR 3,2 ELSE COLOR 1,2
 LOCATE 6+x,20:PRINT names$(x)
 LOCATE 6+x,50:PRINT scores(x)
NEXT x
```

Put the cursor in the correct position and ask the player to input his/her name.

```
COLOR 3,2
LOCATE 6+here,18:INPUT "";nam$
nam$=LEFT$(nam$,20)
names$(here)=nam$
LOCATE 6+here,18:PRINT "                              ":COLOR 3,2
LOCATE 6+here,20:PRINT nam$
LOCATE 6+here,50:PRINT scores(here)
COLOR 1,0
```

Save the new high score file to disk for future reference.

```
OPEN "memory.score" FOR OUTPUT AS #1
FOR x=1 TO 10
 WRITE #1,names$(x),scores(x)
NEXT
CLOSE #1
```

A routine that will pause the program until the player clicks the mouse.

```
LOCATE 20,25:PRINT"Click mouse to continue."
m=MOUSE(0)
WHILE MOUSE(0)=0
WEND
CLS
RETURN
```

This is the routine that draws the game board. I left the ability to cheat in the code. I initially put this in the code so I could debug it. See if you can figure out how to engage 'cheat mode'.

```
board:
m=MOUSE(0): IF MOUSE(1)<5 THEN cheat=1
LINE (50,25)-(590,125),0,bf
LINE (50,25)-(590,125),1,b
LINE (320,25)-(320,125),1
LINE (50,75)-(590,75),1
COLOR 2,0:LOCATE 1,30:PRINT "MemorySquares by Mike Mo"
COLOR 1,0:LOCATE 2,8:PRINT "Score=";sc
COLOR 3,1:LOCATE 2,60:PRINT "Hi-Score=";hs:COLOR 1,0
SOUND 550,5:GOSUB bigpause
LINE (50,25)-(590,125),2,b
LINE (320,25)-(320,125),2
LINE (50,75)-(590,75),2
RETURN
```

Now we see if the player remembers the sequence.

```
turn:
didit=0:dum=d:d=.3
LOCATE 20,35:PRINT"GO!!"

FOR q=1 TO round
 IF cheat THEN LOCATE 10,1:PRINT where(q)
```

See where the player is clicking the mouse. If the player clicks the wrong square then we set the 'didit' flag to 1 (didn't make it), put the round back to the last round, and decrease 'd' so the next time the game goes faster.

```
 GOSUB interpit
 IF pt<>where(q) THEN
  didit=1:round=round-2:d=d+.1
  SOUND 170,10
  LINE (50,25)-(590,125),2
  LINE (50,125)-(590,25),2
```

Decrease the number of 'men' left and then gosub to 'menleft' to see if the game is over.

```
  noleft=noleft-1
  GOSUB menleft
  GOSUB bigpause
  IF noleft=-1 THEN GOTO lp1
 END IF
 IF didit THEN lp1
```

If the correct square was picked, then flash it.

```
 ON pt GOSUB bx1,bx2,bx3,bx4
```

Update the score and check to see if the new score is greater than the current high score. If it is, the high score is updated and printed on the screen.

```
 sc=sc+(10*round):LOCATE 2,8:PRINT "Score=";sc
 IF sc>hs THEN hs=sc:COLOR 3,1:LOCATE 2,60:PRINT "Hi-
Score=";hs:COLOR 1,0
NEXT q

lp1:
LOCATE 20,35:PRINT "    "
d=dum
RETURN
```

A routine that causes a large pause.

```
bigpause:
FOR dd=1 TO 3000:NEXT dd
RETURN
```

Check where the mouse is being clicked and convert it into the number of the square that is there.

```
interpit:
WHILE MOUSE(0)<>0
WEND
WHILE MOUSE(0)=0
WEND
mx=MOUSE(1):my=MOUSE(2)
IF mx<320 THEN
 IF my<75 THEN pt=1 ELSE pt=3
ELSE
 IF my<75 THEN pt=2 ELSE pt=4
END IF
RETURN
```

A graphic and music display that rewards the player for remembering the entire sequence.

```
minicelebrate:
c=2:d=.2
FOR p=1 TO 4
 ON p GOSUB bx1,bx2,bx4,bx3
NEXT p

pause

FOR p=1 TO 4
 ON p GOSUB bx1,bx2,bx4,bx3
NEXT p

d=dum:c=3
GOSUB bigpause
RETURN
```

Erase each 'man' that is used, leaving the number of 'men' that remain displayed.

```
menleft:
FOR ml=noleft+1 TO 3
LINE (10+(ml*60),160)-(60+(ml*60),180),0,bf
NEXT ml
RETURN
```

The next four sub routines draw the four squares and play a sound that is exclusive to each square.

```
bx1:
LINE (51,26)-(319,74),c,bf
SOUND 440,3
pause
LINE (51,26)-(319,74),0,bf
RETURN

bx2:
LINE (322,26)-(589,74),c,bf
SOUND 493.88,3
pause
LINE (322,26)-(589,74),0,bf
RETURN

bx3:
LINE (51,76)-(319,124),c,bf
SOUND 523.25,3
pause
LINE (51,76)-(319,124),0,bf
RETURN

bx4:
LINE (321,76)-(589,124),c,bf
SOUND 587.33,3
pause
LINE (321,76)-(589,124),0,bf
RETURN
```

There is only one sub program used in this program. Sub programs allow us to add commands to AmigaBASIC. Here we add the command 'pause' that delays for a ceratin amount of time depending on the value of 'd'. As you can see, the variable 'd' is SHARED. This means we can set the value of 'd' somewhere else and the sub program can still 'see' it. All other variables used within a sub program that are not SHARED are exclusive to the sub program.

```
SUB pause STATIC
SHARED d
FOR x=1 TO 1000*d:NEXT
END SUB
```

•AC•

This Fall see all the best of the Amiga

AC's GUIDE To The Commodore AMIGA®

When the nights turn crisp and your thoughts return to your Amiga, let AC's Guide To The Commodore Amiga be your guide.

A Harvest of over 2100 products by more than 400 vendors!

During the summer, Amiga developers throughout the world have been diligently preparing Amiga products for all your needs. Now showcased in this, the largest Amiga-specific publication ever, is the harvest of their labors. Your guide will be an indispensable tool to doing more with the Amiga. Don't miss it!

C Notes

from the C Group

Utility programming

by Stephen Kemp

Last month I ended the column with a listing of a sample "skeleton" for many utility programs called "generic.c." One of the things I like about programming in C is that if you need a little program to do some specific task, it usually isn't too difficult to write one yourself. This is especially true if you already have a sufficient number of tools to help out and skeletons from which to start. This month, for your perusal and my discussion, I have included a small utility that was developed from the simple program provided last month.

For those of you who may have missed last month's column, don't despair. The entire source is included, and I will try to point out what is skeletal and what isn't. This should enable you to recreate the basic program, so it can be used in your own utilities. Anyone who did type in last month's column can begin the experiment.by copying the code to a new file. (Note: I recommend working on a copy of the file because you may need to start over, and you may want to start other programs from the same file).

You may recall from past articles that I believe a utility program should have one specific task. Now this does not mean that a program cannot have a multipurpose function, but a program that must support multiple tasks will usually be cumbersome to use. "Keep it simple" is a good motto to follow in utility programming.

Listing 1 is a "case" program—alphabetic case, that is. It simply changes all the alphabetic letters in a text file to upper or lowercase letters. As a teaching aid, this program also demonstrates a few things that newcomers to C should learn. Elaboration on these topics will occur as the listing is discussed.

First note that the comment block at the top of the program was changed from that included in the "generic.c" listing provided last month. It is important to keep your comment blocks up to date. Yeah, I know, you want to get right to the programming. But a few extra minutes of typing may prove useful if you have to look at this code again in a few months, or years. This could be especially true if some other poor soul ends up having to maintain the source. I like to keep a history of revisions under the program comments. A date, initials, and a short sentence or two about the revision can provide a nice audit trail. (If you have not yet realized it, I assume that you are, or one day will be, involved in writing programming with others.)

A few new variables that were not in the original have been declared in this listing. First you will notice two variables declared with the type "FILE", *ifp and *ofp. As you may guess, these are pointers to a file. The FILE type is usually defined (as a typedef) in one of the standard C header files. It defines a structure that maintains information about the status of an input and/or output stream. You may recall that an I/O stream can include the console, printer, or almost any other device or disk file. At this point it is not really necessary for you to know exactly what is contained in the FILE declaration. Just recognize that it is used to accomplish "buffered" input and output.

Standard C libraries normally provide buffered I/O. This method (as opposed to its unbuffered counterpart) means that the system will perform I/O by physical blocksize. A blocksize is that amount of data which is usually the most efficient for the device to move during a given event. Using a disk drive as an example, it is usually more efficient for the drive to read more than one byte in a revolution. The actual blocksize used is usually some multiple of 256 or 512, depending on the device.

So if you are reading from a stream, the first read will fill the buffer regardless of how much data you requested. Subsequent calls do not have to read from the disk again until the buffer is empty. Naturally, if you ask for more data than a buffer can hold, the device will continue transferring data until your request is met. Any data left in the last buffer will be used in the next call. In the case of text files, where lines will be of varying lengths, buffered I/O is especially useful.

Two other variables that were not in the original source have been declared. The first is a "flag" which will determine what case (upper or lower) to apply. The second is simply a character buffer large enough to hold the longest line we expect to read.

The flag setting is expected to be found on the command line. The source to determine this state can be found after the program's "error" exit code. (The exit code should also be kept up to date.) This program assumes that the flag setting can be identified on the command line by a leading hyphen (-) followed by the letter U or L, meaning UPPER or LOWER. The letter following the hyphen is the command of choice and is placed into the variable flag. (Note: When you parse the command line for switches or flags, be sure to "index" into the arguments. If you change the argument pointer, you may interfere with references to that variable which occur later in the program.)

Also note that this program loops through all the arguments and, if multiple flags are discovered, only the last flag takes effect. This is something you can handle any way that you want. You might say this would be an error, or you could even change the program to use one flag for a while and then change in midstream. To simplify, this program will use the same setting throughout. After the arguments are examined, the flag is changed to an uppercase letter to reduce the code requirement later in the program. However, if the flag has not been set or is set incorrectly after each argument is examined, control is sent to the error exit.

The next new code occurs after a file name that matches the command line is found. When starting from the generic.c source, this will be where most of the utility's code will be found. Notice that the file is opened twice—once for reading and once for update.

Now the question on your mind is, "Why open the file twice?" Well, we could have opened the file once, but when you are using buffered I/O to do a job like this, it can be a little tricky. The reason is that this program changes the characters "in place." It is not writing a new file, but rather expects to place the buffer (after alteration) right back where it was found. Buffered streams do not provide a straightforward method to accomplish this task. To draw an analogy, think of a river stream. Streams flow in one direction. The same is basically true for these buffered streams, which means that extra code is required to help keep the direction identified. Now I don't want it to sound worse than it really is. It doesn't require a lot of code. It's just that the source provided here is more straightforward. There is no problem in opening the file twice like this. The first file pointer keeps track of the reading, and the second the writing. Since we are putting back exactly what we have read, the two file pointers will not conflict.

The actual case change is handled by one of two functions provided at the bottom of the listing. These are fairly simple functions that can usually be found in the standard library provided with many compilers. Manx, however, did not include these two, so I wrote the functions myself. Each uses a pointer to move through a string, altering each character by calling another standard function.

When you put it all together, compile and link it, you should have a working program. Test it out on a few documents. DO NOT use it on your source code or you'll be sorry. Remember, C is case-sensitive. Try it on other documents or batch files, etc.

That's it. CASE will alter the alphabetic letters in each file that matches the command line criteria. If you have a little extra time and energy, take the program and try to make a couple of your own modifications. One possibility I mentioned earlier would be to switch flags in midstream when a new flag is encountered. Another would be to assume some default flag. Let's face it, if you are changing the case of an entire file, what is the worse that could happen? If you get the case wrong, just run it again with the correct flag. Try a few experiments. You never know what you might discover!

We take a byte out of the price

ONE BYTE

P.O. Box 455
Quaker Hill, CT 06375
(800) 441-BYTE, in CT (203) 443-4623

Authorized dealer for
Commodore-Amiga Computers,
Great Valley Products (GVP),
Memory & Storage Technology (M.A.S.T.).
Authorized Commodore-Amiga Service and Repair.
Authorized Amiga Graphics Dealer.

Circle 129 on Reader Service card.

AMIGA IS A REGISTERED TRADEMARK OF COMMODORE-AMIGA, INC.

Listing One: CASE.C

```
/* CASE.C is a program that accepts a list of filenames or wild- */
/* cards and will read and rewrite each program altering the     */
/* case.  The state of the case is determined by a command line  */
/* switch.  -U indicates that the files should be changed to     */
/* upper case.  -L indicated that the files should be changed to */
/* lower case.  Only one flag is used for all files.             */
/* An example command line might look like this:                 */
/*      CASE -u *.txt file.c docs.*                              */
/* If a “standard” open fails on a file it is skipped.           */
/* This program was written for the MANX C compiler              */

#include “stdio.h”
#include “fcntl.h”

extern char   *scdir();                /* directory function*/
extern short  access();                /* file accessable function*/

main(argc,argv)                        /* program start */
short argc;                            /* argument counter */
char *argv[];                          /* argument variable pointer(s)*/
{
    char *fptr;                        /* pointer to a filename*/
    short cnt;                         /* counter for files     */
    short ndx;                         /* index for arguments   */
    char  flag;                        /* case flag             */
    FILE  *ifp,*ofp;                   /* file io pointers      */
    char  lbuf[256];                   /* line buffer           */

    if (argc <= 2){                 /* if not enough arguments provided */

errout:                                /* exit point if error discovered */

       printf(“Change the alphabetic case of a file \n”);/* announce */
       printf(“   CASE -flag  [file pattern] ...\n”); /* example */
       printf(“     where flag is:   -U for upper case\n”); /* upper */
       printf(“                 or:   -L for lower case\n”); /* lower */
```

```
        exit(0);                        /* exit the program */
    }

/*——— Check for flags and switches here ——————————--------------*/

    flag = '\0';                                  /* indicate no flag*/

    for(ndx = 1; ndx < argc ;ndx++){              /* examine arguments*/
        if (argv[ndx][0] == '-')                  /* found an indicator*/
            flag = argv[ndx][1];                  /* store the flag */
    }

    flag = toupper(flag);                          /* alter case*/
    if (flag != 'U' && flag != 'L') goto errout;
                                               /* no indicators - exit*/

    printf("Convert to %s case \n",((flag == 'U')? "UPPER":"LOWER"));

/*——————————————————————————————-----------------------------------*/

    for(ndx = 1; ndx < argc ; ndx++){            /* examine arguments*/

        if (argv[ndx][0] == '-')                 /* flags already found*/
            continue;                            /* so skip to next */

         for(cnt=0;(fptr=scdir(argv[ndx]));cnt++){
                                                /* look for wildcards */
            if (access(fptr,0) == 0){            /* if file is found */
                printf("*** %s ***\n",fptr); /* display filename */

/*——  Utility program specific code goes here ————————-----------*/

                ifp = fopen(fptr,"r");      /* open file for read*/
                ofp = fopen(fptr,"r+");     /* open file for update*/
                if (ifp == NULL || ofp == NULL){ /* if opens fail*/
                    printf(" - skipped - \n");
                    if (ifp) fclose(ifp);       /* close if open */
                    if (ofp) fclose(ofp);       /* close if open */
                    continue;                   /* continue looping*/
                }
                for(;;) {                      /* loop through file */
                    if (fgets(lbuf,256,ifp) == NULL) { /* EOF */
                        fclose(ifp);            /* close file */
                        fclose(ofp);            /* close again*/
                        break;                  /* end loop */
                    }
                    if (flag == 'U')            /* if upper flag */
                        strupr(lbuf);           /* up the string */
                    else                        /* otherwise */
                        strlwr(lbuf);           /* lower the string*/
                    fputs(lbuf,ofp);            /* put the line back*/
                }

/*————————————————————————————------------------------------*/
            }else{                                /* if not found */
                printf("Cannot find %s\n",fptr);/* indicate */
            }
        }
        if (cnt == 0)                             /* no matches */
            printf("Cannot find %s\n",argv[ndx]);/* for this argv */
    }
}

/* strupr is a function that indexes through a string and converts */
/* all alphabetic characters to upper case */
strupr( str )
char *str;
{
    for( ;*str != '\0'; str++)     /* search until null is found */
        *str = toupper(*str);      /* call upper case function */
}

/* strlwr is a function that indexes through a string and converts */
/* all alphabetic characters to lower case */
strlwr( str )
char *str;
{
    for( ;*str != '\0'; str++)     /* search until null if found */
        *str = tolower(*str);      /* call lower case function */
}
```

•AC•

July 1989 Volume 4 Number 7

Did you miss this special Amazing Issue?

Don't miss another issue, Subscribe today!

Back issues are available while supplies last.

AC EXCLUSIVE!

Amiga's version of Apple's HyperCard With COLOR!

An Inside look at UltraCard by Steve Gillmor
A behind the scenes look at the Amiga's first *HyperCard* style program.
Adapting Analog Joysticks to the Amiga by David Kinzer
David Kinzer shows Amiga game buffs how to arm their machine with an analog joystick.
C Notes from the C Group by Stephen Kemp
Good planning for good programming.
Using Coordinate Systems by Paul Castonguay
Part II of the Fractals series addresses the basis of computer graphics.
1989 Spring Comdex, Chicago by Ed Bercovitz

AC EXAMINES AMIGA GAMES!

Wayne Gretzky Hockey review by Bob Borgen
Is the game as great as the name?
Western Games review by Stephen Kemp
Quid spittin', cow milkin', beer shootin', and watch those belches.
The Duel review by Joe DiCara
Joe blasts down the highway in this sequel to Test Drive.
Baal review by Derek J. Perry
Derek straps on the gear and goes after the war machine.
DataStorm review by Paul Costa
Defender's mutants invade the Amiga.
Lords of the Rising Sun review by Derek J. Perry
Samurai fighting in Shogun Japan.
Dungeon Master review by Graham Kinsey
Enter the dark halls of Lord Chaos' dungeon.
Zany Golf review by Joe DiCara
Plaid pants aren't the only crazy thing on this golf course.
Hole-in-One Miniature Golf review by Stephen Kemp
Putt putt will never be the same.
Deja Vu review by Bruce Jordan Have I done this before???
Battle Chess review by Jeffery Scott Hall
An action-packed version of an old classic.
Snapshot by R. Bradley Andrews
Three arcade games, and a day at the lake.
Amazing Amiga Games Listing
Over 450 Amiga entertainment products listed!

AC MONTHLY COLUMNS!

New Products...and Other Neat Stuff by Elizabeth G. Fedorzyn
Mac emulator, Electric Crayon, & Where Is Carmen Sandiego?
Ask AC AC's tech editor answers your questions.
The Video Desk by Larry White
Larry looks at where Amiga video has been and where it's going.
PD Serendipity Fred Fish 211 to 220, latest in software philanthropy.
Bug Bytes by John Steiner Upgrades from Gold Disk, B.E.S.T. & more.
Roomers by The Bandito More personnel changes at CBM.
HardCopy by Melissa J. Bernier and Paul Costa AC's new games page.
HyperCard is a trademark of Apple Computer, Inc.

Improving Your Graphics Programming

by Richard A. Martin

You work for hours on a hot new program for your Amiga, slaving over every graphic detail. All your gadget imagery is immaculate, your IntuiText tasteful. It's just beautiful. You're so thrilled with your work, you call up a fellow Amigoid and invite yourself over to show off your efforts. When you arrive, you eagerly insert your disk into your host's drive and begin to run your program.

But what's this? Your fellow Amigoid's Workbench colors are entirely different from your own. Since your program opens its window on the Workbench screen, it uses those colors selected by Preferences. The program that looked great on your screen at home now looks dreadful on someone else's Amiga. As you try to explain this to your now skeptical (and unimpressed) Amigoid host, you realize that you have just been victimized by "Blind Artist's Syndrome."

The Blind Artist draws very well, but it doesn't know color from a hole in the ground. It will draw with any old color, paying no heed to the fundamentals of light and shadow. The Blind Artist reaches out, grabs pen number three, and draws like crazy, caring nothing about which color ink the pen contains. Not surprisingly, nobody buys the Blind Artist's drawings.

If you write programs which open their windows on the Workbench screen (or on any other screen which you did not open yourself), you run the risk of contracting Blind Artist's Syndrome each time you use arbitrary pen numbers to do your graphics rendering. Since each user has the option of choosing a unique set of Workbench colors, you, as the programmer, make a rash assumption if you presume the default Preferences palette is unchanged. It's a pretty safe bet that using color 0 for background and color 1 for line drawing will result in a readable display. But this is an Amiga, not a Mac. With four colors available on the Workbench screen, your renderings can be enhanced by highlighting important items, subduing lesser material, and shadowing graphics with the appropriate pens. Simple #defines such as

```
#define MED_BLUE    0
#define WHITE       1
#define BLACK       2
#define ORANGE      3
```

may work fine on a factory-fresh Workbench disk, but out there in the real world, many users are abandoning that hideous color scheme left and right.

If you're like me, you spend a considerable amount of time laboring over attractive display graphics with such "3D" elements as drop-shadowed text and realistic looking buttons. Personally, I prefer to make color 1 BLACK and color 2 WHITE. As a result, many programs look odd on my Amiga, with highlights appearing where shadows ought to be. This technique gives the display a sort of Bela Lugosi-style -- "lit-from-the-bottom" look.

You can avoid Blind Artist's Syndrome by tailoring your graphics routines to utilize the current Workbench colors more intelligently. Once you know which colors you have to work with, you can choose those most appropriate for your graphic purposes. Here is one method for choosing the most appropriate pen, no matter which four colors are being used at the time.

How it all works

Since the Workbench screen employs only four colors, I sort them from darkest to lightest, giving them generic names like Shadow, Subdue, BackGround, and Hilite. I keep these names handy by creating the following global structure of the Palette type, which I name WB_Palette:

```
struct Palette {
BYTE Shadow;
BYTE Subdue;
BYTE BackGround;
BYTE Hilite;
} WB_Palette;
```

Each of the four elements contains the current Workbench pen numbers that best suit the color value. The Shadow element contains the pen number of the darkest color currently displayed on the Workbench, Subdue the second darkest pen number, and BackGround the third, leaving Hilite for the brightest available pen. By selecting the darkest and brightest pens available (Shadow and Hilite, respectively), I can render crisp, easy-to-read text by printing a Hilite color over a Shadow color, at a slight offset.

The nitty gritty

Intuition provides a function named GetPrefs() which allows you to examine all the settings you have made with the Preferences tool. A call to GetPrefs() will yield an abundance of user-defined data, but the focus here is on only a small portion of the Preferences data structure. Nearly halfway through the list of settings are four 16-bit elements named color0, color1, color2, and color3. Each of these contains the Red, Green, and Blue values of the current Workbench pens. By sorting these colors from darkest to lightest, you can be sure that your shadows are dark, your window background is muted, and your text is bright and easy to read. A call to GetPrefs() takes the following form.

```
GetPrefs(PrefBuffer, Size);
```

PrefBuffer is a pointer to a memory buffer that you want filled with Preferences data, and Size is the number of bytes you want filled in. It is up to you to allocate the memory needed to hold the Preferences data. Intuition groups the more common settings at the top of the Preferences structure, so reading the whole thing is rarely necessary. For example, to check whether the user has selected 60 or 80-column text mode, simply call GetPrefs() with a Size argument of 1L, since FontHeight is the first element in the Preferences structure. When it returns, your little one-byte buffer will contain a value of either TOPAZ_SIXTY or TOPAZ_EIGHTY, depending on the current setting. In the case at hand, you will read 118 bytes of data, just enough to get the current colors.

To hold your color data, you must allocate memory for an array of four Color structures. Each Color structure contains four elements, each of which is 4 bits in size. You then utilize a special feature of the C language known as a bitfield to hold each nybble-sized value. Using bitfields, you can access tiny objects packed into a single word. Intuition packs the Blue, Green, and Red values into the three least significant nybbles of an unsigned SHORT. You will want to take advantage of the unused fourth nybble for your own purposes.

```
struct Color {
unsigned Blue  : 4;
unsigned Green : 4;
unsigned Red   : 4;
unsigned Value : 4;
};
```

Bitfields look and behave much the same as ordinary structure elements, with a few differences. Most obvious is the inclusion of a colon in the element definition, followed by the field size, expressed as the number of adjacent bits. Also notice that the fields are listed with the least significant bits at the top of the list. This is done because the Amiga assigns bitfields from right to left. On other computers, they might be assigned from left to right, so bitfields do not lend themselves well to portable code. Also, since each field is only a portion of an int, you cannot take the address of a bitfield. However, in other respects you can treat bitfields as though they were regular integers.

Once you assign the RGB values of your Color structure, pass them to getValue() which makes a rough approximation of the color's brightness by averaging the Red, Green, and Blue values of the Color in question. The result is then placed in the most significant nybble of the Color structure. This averaging algorithm is an oversimplification of actual perceived brightness: it discards any R, G, or B value less than four (values this low contribute almost nothing to the color's brightness) and takes an average by integer division. With most Workbench color schemes, it seems to do the job. When I tested it on some more exotic palettes, however, the color assignments came out less than perfect. (Incidentally, anyone who cares to improve the reliability of this function is heartily encouraged to do so. Kindly send me a copy when you are done.)

Your freshly assigned Color structure is actually the first element of a slightly larger structure I call a Color_Register. The second element is simply a SHORT integer used for storing the pen number. Since you will be shuffling these structures, you will need to know the original pen numbers when it comes time to do some actual drawing. Once your Color_Register structures are completely filled out, you can free up the memory allocated for the Preferences data and proceed to the next step: sorting the colors. sortWBenchValues() is only a slight modification of the shell sort routine found in Chapter 3 of The C Programming Language by Kernighan & Ritchie. I simply customized it to handle the specific task of sorting an array of four Color_Register structures from darkest to lightest, according to the cr_Color.Value element. Once this sort is done, just copy the original pen numbers into your Palette structure, deallocate your array of Color_Registers, and begin drawing.

Putting it to use

Once your WB_Palette is initialized, you can render freely in your window, reasonably confident that your images will look as good as they can given the available colors. If you desire a dark color, you can specify WB_Palette.Shadow or WB_Palette.Subdue. You can utilize the brightest color by specifying WB_Palette.Hilite. The final chapter of *Style,* the Intuition manual, mentions that Intuition's Pointer is designed with the light source coming from the upper right. To uphold this convention, I have assigned color values in the demo program accordingly, with shadows appearing at the lower left. Note how this can be used to create the illusion of both raised and recessed surfaces within your window.

Drop-shadowed text can be rendered easily using a linked pair of IntuiText structures. First, draw your text using the WB_Palette's Shadow color for the FrontPen and the WB_Palette's BackGround color for the BackPen. In Drawing Mode JAM2, these settings will both erase any existing background and draw the text. Next, draw your text again using the WB_Palette's Hilite color—not directly on top of our shadow, but offset slightly upwards and to the right. This time, use Drawing Mode JAM1 so only the letters will be drawn over the shadow.

The following sample program opens a simple Workbench window, displays some text and graphics using the WB_Palette method, and waits for either a mouse click in its Close gadget or any keypress before exiting. Note that the IDCMPFlags of this window include the NEWPREFS bit. If, during the course of the program's execution, you choose to alter Workbench's Preferences structure, Intuition will send an IntuiMessage of class NEWPREFS. You can then reexamine the current colors and redraw your graphics accordingly. You can experiment by running SortColors and Preferences simultaneously, watching SortColors redraw its window after each invocation of Preferences.

If you are satisfied that your graphics programming is safe even in the hands of the most colorblind Workbench user, you can include sort_colors.h and sort_colors.c in your own programs. All you need to do is call ChoosePens() after opening the intuition.library and before you begin rendering in your window. If Intuition is doing any amount of rendering for you—Gadget imagery, Menus, or IntuiText—you will probably want to call a routine which initializes those colors before you open your window. With a WB_Palette handy, impressive graphics are automatically managed for any Workbench color scheme.

Listing One: sort_colors.h

```
/*************************************************/
/* sort_colors.h */

#ifndef SORT_COLORS_H
```

Amazing on Disk

Source Listings and Executables from the pages of Amazing Computing

Only $6.00 per disk ($7.00 for Non-Subscribers)

Amazing Computing AC #3

Amazing Computing AC #4

See the complete listing on page 26.

```
#define SORT_COLORS_H

#ifndef EXEC_TYPES_H
        #include <exec/types.h>
#endif EXEC_TYPES_H

struct Color {                /* we use bit-fields here */
        unsigned Blue  : 4;    /* because that's the way */
        unsigned Green : 4;    /* Intuition stores       */
        unsigned Red   : 4;    /* the color data         */
        unsigned Value : 4;    /* use last 4 bits        */
};
/* for brightness */

struct Color_Register {
        struct Color    cr_Color;
        SHORT           cr_Number;
};

struct Palette {        /* we use this to store the     */
        BYTE Shadow;    /* color register numbers which */
        BYTE Subdue;    /* correspond to our shading    */
        BYTE BackGround; /* scheme.                     */
        BYTE Hilite;
};

/* this is our global Palette structure, */
/* used in all our display routines */
extern struct Palette WB_Palette;

#endif SORT_COLORS_H
```

Listing Two: sort_colors.c

```
/**********************************************/
/* sort_colors.c */

#ifndef SORT_COLORS_H
        #include "sort_colors.h"
#endif SORT_COLORS_H

#ifndef EXEC_MEMORY_H
        #include <exec/memory.h>
#endif EXEC_MEMORY_H

#define OUTTA_MEMORY        1
#define SUCCESS             0

struct Palette WB_Palette;

/* this guy looks at the current WorkBench colors,     */
/* computes the brightness value of each, sorts        */
/* them from darkest to lightest, and puts the proper */
/* pen numbers into a global Palette struct.           */
ChoosePens()
{
#define RED_PEN             (reg_buf+i)->cr_Color.Red
#define GREEN_PEN           (reg_buf+i)->cr_Color.Green
#define BLUE_PEN            (reg_buf+i)->cr_Color.Blue

        SHORT i;
        SHORT *pref_buf;
        struct Color_Register *reg_buf;
        void *AllocMem();

        if(!(pref_buf = (SHORT *)AllocMem(118L, MEMF_FAST))) {
                return(OUTTA_MEMORY);
        }
        if (!(reg_buf = (struct Color_Register *)
        AllocMem(sizeof(struct Color_Register) *4L,
                        MEMF_FAST))) {
                FreeMem(pref_buf, 118L);
                return (OUTTA_MEMORY);
        }

        /* we only need the first 118 bytes */
        /* to find the color data           */
        GetPrefs(pref_buf, 118L);

        for (i=0; i<4; i++) {
                (reg_buf+i)->cr_Color  = *(pref_buf + 55 + i);
                (reg_buf+i)->cr_Color.Value =

                getValue(RED_PEN,GREEN_PEN,BLUE_PEN);
                        /* save the pen number */
                (reg_buf+i)->cr_Number = i;
        }

        FreeMem(pref_buf, 118L);

                /* sort them from darkest to lightest */
        sortWBenchValues(reg_buf);

        WB_Palette.Shadow        = (reg_buf  )->cr_Number;
        WB_Palette.Subdue        = (reg_buf+1)->cr_Number;
        WB_Palette.BackGround    = (reg_buf+2)->cr_Number;
        WB_Palette.Hilite        = (reg_buf+3)->cr_Number;

        FreeMem(reg_buf, sizeof(struct Color_Register) * 4L);

        return (SUCCESS);

}       /* end ChoosePens() */

/***********************************************/
/* This routine could stand some improvement. */
/* A true evaluation of how the eye */
/* perceives brightness is beyond me. */
/* So for now, we just take an average */

getValue(r,g,b)
WORD r, g, b;
{
#define AVG_VALUE(a,b,c)  (((a)+(b)+(c))/3)

        r = r<4 ? 0 : r;
        g = g<4 ? 0 : g;
        b = b<4 ? 0 : b;

        return (AVG_VALUE(r,g,b));
}

/***************************************************/
/* This is a modification of the shell() function */
/* listed in K & R. It specifically sorts 4 */
/* Color Register structures */

sortWBenchValues(v)
struct Color_Register v[];
{
#define VAL               cr_Color.Value

        WORD gap, i, j;
        struct Color_Register temp;

        for (gap = 2; gap > 0; gap /= 2)
        for ( i = gap; i < 4; i++)
                for (j = i-gap;
                        j >= 0 && (v[j].VAL > v[j+gap].VAL);

                                j -= gap) {
                                temp = v[j];
                                v[j] = v[j+gap];
                                v[j+gap] = temp;
                        }
}
```

Listing Three: demo.c

```
/***********************************************/
/* demo.c */

#ifndef INTUITION_INTUITION_H
        #include <intuition/intuition.h>
```

About the Author

Rich Martin graduated from U.C.L.A. in 1978 with a B.A. in Motion Pictures/Television. A musician all his life, he specializes in MIDI and electronic music and has been recording for the past decade. With a knowledge of C, BASIC, and AmigaDOS, Rich has been able to use his Amiga as a powerful tool in making music.

Rich's passion for animation is almost as old as his dedication to music. When free from his duties as Amiga manager of Do_While Studios, a computer arts cooperative in Boston, he indulges himself with the help of Deluxe Video, Sculpt-Animate-3D, and DeluxePaint III.

```
#endif INTUITION_INTUITION_H

#ifndef SORT_COLORS_H
        #include “sort_colors.c”
#endif SORT_COLORS_H

/* first, a couple of IntuiTexts */
/* for drop-shadowed text */
struct IntuiText text[] = {
        { 1, 0, JAM2, 0,  0, NULL, NULL, &text[1] },
        { 1, 0, JAM1, 2, -1, NULL, NULL, NULL },
};

/* a few drawing vectors */

#define BUTTON_WIDTH        48
#define BUTTON_HEIGHT       15

#define TITLE_WIDTH         300
#define TITLE_HEIGHT        30

WORD button_hi_vects[] = {
        1, 0,
        BUTTON_WIDTH-1, 0,
        BUTTON_WIDTH-1, BUTTON_HEIGHT-2,
        BUTTON_WIDTH-2, BUTTON_HEIGHT-2,
        BUTTON_WIDTH-2, 1
};

WORD button_sh_vects[] = {
        BUTTON_WIDTH-1, BUTTON_HEIGHT-1,
        0, BUTTON_HEIGHT-1,
        0, 0,
        1, 1,
        1, BUTTON_HEIGHT-2
};

WORD title_hi_vects[] = {
        1, 0,
        TITLE_WIDTH-1, 0,
        TITLE_WIDTH-1, TITLE_HEIGHT-2,
        TITLE_WIDTH-2, TITLE_HEIGHT-2,
        TITLE_WIDTH-2, 1
};

WORD title_sh_vects[] = {
        TITLE_WIDTH-1, TITLE_HEIGHT-1,
        0, TITLE_HEIGHT-1,
        0, 0,
        1, 1,
        1, TITLE_HEIGHT-2
};

struct Border border[] = {
{               /* these two get the raised look */
        0, 0,                                   /* Left, Top */
        0, 0, JAM1,                             /* pens, mode */
        5, button_hi_vects,
        &border[1] },                 /* next */
{
        0, 0,                                   /* Left, Top */
        0, 0, JAM1,                             /* pens, mode */
        5, button_sh_vects,
        NULL },
{               /* these two get the recessed look */
        10, 13,                                 /* Left, Top */
        0, 0, JAM1,                             /* pens, mode */
        5, title_sh_vects,
        &border[3] },                 /* next */
{
        10, 13,                                 /* Left, Top */
        0, 0, JAM1,                             /* pens, mode */
        5, title_hi_vects,
        NULL },
};

#define WINDOW_FLAGS        WINDOWCLOSE | ACTIVATE |\
                            WINDOWDRAG | WINDOWDEPTH |\
                            SMART_REFRESH


TEXT window_title[] = {“ Window Sweet Window ”};

struct NewWindow new_window = {
        160, 50,                      /* Left, Top     */
        320, 100,                     /* Width, Height */
        -1, -1,                       /* Pens          */
        CLOSEWINDOW | RAWKEY | NEWPREFS,
        WINDOW_FLAGS,
        NULL,                                   /* 1st Gadget */
        NULL,                                   /* CheckMark  */
        window_title,
        NULL,                                   /* Screen     */
        NULL,                                   /* BitMap     */
        0, 0,
        320, 100,
        WBENCHSCREEN
};

struct Library    *IntuitionBase;
struct Library    *GfxBase, *OpenLibrary();

struct Window     *window, *OpenWindow();


char *strings[] = {
        “’Twas brillig and the slithy toves,”,
        “ Did gyre and gymble in the wabe.”,
};

DoText(rp)
struct RastPort *rp;
{
                /* set up the colors */
        text[0].FrontPen = WB_Palette.Shadow;
        text[0].BackPen = WB_Palette.BackGround;
        text[1].FrontPen = WB_Palette.Hilite;

                /* plug in the first line */
        text[0].IText = text[1].IText = (UBYTE *)strings[0];
        PrintIText(rp, text, 18L, 20L);

                /* plug in the second line */
        text[0].IText = text[1].IText = (UBYTE *)strings[1];
        PrintIText(rp, text, 18L, 30L);
}

DoBoxes(rp)
struct RastPort *rp;

{
        LONG x, y;
                /* set up the colors */
        for (x=0L; x<4L; x+=2L) {
                border[x  ].FrontPen = WB_Palette.Hilite;
                border[x+1].FrontPen = WB_Palette.Shadow;
        }

                /* draw the title box */
        DrawBorder(rp, &border[2], 0L, 0L);

                /* and a bunch of fake buttons */
        for (x=36L; x<320L - BUTTON_WIDTH; x+=50L)
                for (y=46L; y<100L - BUTTON_HEIGHT; y+=16L)
                        DrawBorder(rp, border, x, y);
}
```

```
DoBackGround(win)
struct Window *win;
{
        /* Gotta fill the background if */
        /* color 0 is not acceptable */
        if (WB_Palette.BackGround != 0) {
            SetAPen(win->RPort,(long)WB_Palette.BackGround);
            RectFill(win->RPort, 2L, 10L,window->Width-3L,
                        win->Height-11L);
            RectFill(win->RPort, 2L, (long)win->GZZHeight,
                        win->GZZWidth+3L, win->Height-2L);
        }
}


DrawSome(win)
struct Window *win;
{
        struct RastPort *rp = win->RPort;

        DoText(rp);
        DoBoxes(rp);
}

DoATrick(win)
struct Window *win;
{
        struct RastPort *rp = win->RPort;
        LONG x, y;

        for (x=0L; x<36L; x++) {
                WaitTOF();
                SetRast(rp, (long)(WB_Palette.Shadow));
                WaitTOF();
                SetRast(rp, (long)(WB_Palette.Hilite));
        }
}

OpenShop()
{

        if (NOT(IntuitionBase =
        ReplyMsg(message);

}

PunchOut(code)

WORD code;
{
        if (window)
                CloseWindow(window);
        if (GfxBase)
                CloseLibrary(GfxBase);
        if (IntuitionBase)
                CloseLibrary(IntuitionBase);
        exit(code);
}

main()
{
        struct IntuiMessage *im, *GetMsg();

        OpenShop();
        FOREVER {
                        /* sleep 'til Intuition comes a-
callin' */
                WaitPort(window->UserPort);
                        /* get all the latest news */
                while(im = GetMsg(window->UserPort)) {
                        if (ReadMail(im) == FALSE) {
                                TidyUp(window);
                                PunchOut(0);
                        }
                }
        }
}
```

•AC•

NEW—F-BASIC V2.0—NEW

F-BASIC™ is an Enhanced, Compiled Basic Language System.

★Easy For Beginners
★SO Fast—Beats C!
★Recursive Subprograms
★Over 350 Powerful Commands
★Easy Access to Amiga Libraries
★Comprehensive Users Manual
★Easy Access to Amiga Windows, Menus, Sound, Events & Animation
★Complete For Serious Programmers
★Direct IFF File Support
★Record Structures and Pointers
★Ultra Fast Floating Point
★Sample Program Disk 100 Examples
★Snobol-Like Pattern Matching

The F-BASIC™ System Also Has A Source Level Debugger

★Debug F-BASIC™ Programs at the Source Level—So Easy.
★Intuition Interface—Windows for Source, Registers, Memory, Variables.
★Comprehensive Users Manual—Full Documentation.

The F-BASIC™ Language System 2.0—Only $89.95
F-BASIC™ System and Complete SLDB—Only $149.95

Available Only From The Manufacturer!
SEND CHECK OR MONEY ORDER TO:
DELPHI NOETIC SYSTEMS, INC.
Post Office Box 7722
Rapid City, South Dakota 57709-7722
Credit Card or C.O.D. Call (605) 348-0791

F-BASIC and FastCom are registered trademarks of DNS, Inc.
AMIGA is a registered trademark of Commodore/AIMGA Inc.

Circle 116 on Reader Service card.

MOVING?

SUBSCRIPTION PROBLEMS?

Please don't forget to let us know.
If you are having a problem with your subscription or if you are planning to move, please write to:

Amazing.Computing Subscription Questions
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722

Please remember, we cannot mail your magazine to you if we do not know where you are.

Please allow four to six weeks for processing.

INSTANT GEL-O

Gels In Multi-Forth, Part III

by John Bushakra

I usually make this tasty dish in the morning while I vacuum:

Ingredients:
10 3.5-inch blank, floppy diskettes
1 bottle of tabasco sauce
1 roll of Peppermint LifeSavers
Several pieces of broken glass
1 bottle of fine wine
1 dozen carrots
1 piece of burnt toast
1 egg, rotten
1 ball peen hammer

Place all ingredients except disks, LifeSavers, wine and hammer on pastry board. While drinking wine, pound on board vigorously with hammer until all ingredients are thoroughly pulverized. Pour powdered debris into blender, including hammer for good measure. Set blender to "purée" and let run for 3 minutes. After blending, pour contents into one of your disk drives. Insert disks one at a time and type

```
FORMAT DRIVE DF1: NAME JUNK-FOOD NOICONS
```

Place formatted disks on silver platter, garnish with LifeSavers, and serve chilled.

When listing my faults as a programmer, I have to rank laziness right up there in the top five. The whole Gels In Multi-Forth project was conceived out of laziness. I just cannot stand typing in image bitmaps by hand. Even if you use the public domain IFF routines, you've still got to do all that terrible initialization the hard way—by hand. Anyone who has ever had to do this will tell you that it takes inordinate amounts of time. The resulting code is not only ugly but wasteful, as it is typically executed only once by the main program.

After reflecting upon the problem for a while, I determined that nearly all Gel initialization could be done through a single function, if the necessary information were stored in a disk file. The information was already there; all I had to do was convert it into a usable form—IFF files.

I immediately began writing a program to convert IFF files made with *DPaint II* into files I could read in from the disk, storing the information directly in Gel structures. The result appeared in Part One of Gels In Multi-Forth, which appeared in the April, 1988 issue of *Amazing Computing.*

However, even that program proved to be too hard to use. I knew it had outlived its usefulness after some late-night hallucinations, in which the program asked me to delete its executable image, as well as its own source code. The time had come for me to rewrite the program, using the ultimate lazy-man's user interface: Intuition.

For those uninitiated folks who may be unfamiliar with exactly what the program does, here is a short explanation. The converter program allows the user to draw Gels using one of the myriad paint programs available. The image is then saved to disk as an IFF Brush file. Using the converter, the user may enter information pertinent to Gels—such as x- and y-positions, collision masks, all the various Bob flags, and the zillions of animation object flags—so that it may be stored on a disk. The program then converts the IFF image to the format needed by the Amiga when it is time to display the Gel (see part one of Gels for a detailed explanation of this process). Using specialized functions, these numbers, flags, and converted image data can be read directly back into Gel structures at run time. This is much easier than having to store all the information by hard-coding.

The file format used by the old program proved to be too clumsy for larger programs, so it was also redesigned. Now, the converter program stores more than one Gel in a file, so an entire animation sequence can be read in one shot. Below are the layouts for the new files, beginning with the layout for a hardware sprite:

1.) Number of Gels stored in the file: a long integer (4 bytes).
2.) SimpleSprite structure (defined in graphics/sprite.f).
3.) Sprite image data. Note that the converter program automatically pads the first and last two words of image data with zeroes.

The two blocks above are repeated for each sprite stored in the file. For Virtual Sprites, the file structure is as follows:

1.) Number of Gels in the file (4 bytes).
2.) Virtual Sprite structure (defined in graphics/gels.f Sprite Image Data).

The Virtual Sprite structure and its corresponding image data are repeated for each Gel in the file.

Next, the file format for Bobs:

1.) Number of Gels in the file (4 bytes).

2.) Virtual Sprite Structure.
3.) Bob structure.
4.) Bob image data.

Once again, the Virtual Sprite, Bob, and image data are repeated for each Gel in the file.

For Animation Objects, the file format is as follows:

1.) Number of Gels (4 bytes).
2.) Animation Object structure.
3.) Virtual Sprite structure.
4.) Bob structure.
5.) AnimComp structure.
6.) Image data.

The Virtual Sprite, Bob, AnimComp and image data blocks are repeated for each animation component in the file. Note that the Animation Object structure is not repeated, since only one such structure is needed.

There is no colormap information stored in the Gel files. The converter program still writes out the colormaps to a separate file.

Now take a look at Listing One, which is called "gels.f." This file contains all the Forth code required to read in files made with the converter program. For non-Forth readers, I've included Listing Three, "gels.c," which contains the functions written in C. Listing Two is a short demo program that contains Forth code for placing a hardware sprite, virtual sprite, bob, and a joystick-controlled animation object on the screen.

The first items in "gels.f" are global variables, which are used by all the functions. Why not just include them as "locals"? That was my original intention, but when I got around to writing the function to make an animation object, I ran into some serious trouble. Multi-Forth allows the use of only 10 local variables inside a definition. The "makeaob" function was using 15, so I had to factor these out and make them global.

One of these global variables, "gacarr" (global AnimComp array), is a two-dimensional array that will hold animation components as they are read in from disk. After they are read in, they will be linked together (more on that later).

The first function, "readygels," initializes the Amiga's Gel system and is lifted pretty much intact from its C counterpart in the ROM Kernel Manual. The Gel system keeps track of things using (what else?) a linked list. In this case, it is a list of virtual sprites. In order to use the system, you must allocate dummy head and tail Virtual Sprite structures. All virtual sprites will be inserted into the list between these two.

Another necessary structure is called GelsInfo. The leftmost, rightmost, topmost and bottommost fields in this structure are most important. These fields define the size of the screen for collision-detection purposes.

Also of interest is the "sprRsrvd" field. This field reserves certain hardware sprites, thus making them unavailable for allocation. Experience has shown that these hardware sprite functions should be reserved in a manner exactly opposite to that described in the ROM Kernel Manual.

There are 8 hardware sprites available for use, each of them having a corresponding bit in the byte-sized field called "sprRsrvd." For example, bit 0 corresponds to hardware sprite 0 (the Intuition pointer), and bit 7 corresponds to sprite 7. The manual suggests that storing a 1 in a bit position will reserve the respective sprite. The manual goes on to suggest that storing a 3 in this field will reserve sprites 0 and 1. In fact, the opposite is true. Storing a 0 in the bit position will reserve the sprite. Thus, you must store 252 to reserve sprites 0 and 1 (11111100 binary = 252 decimal).

Notice that all dynamic memory allocation is done using Intuition's AllocRemember function. "Remembering" solves the problem of having scads of ugly memory pointers lying around which you'll have to free up one at a time when you've finished. For arguments, AllocRemember wants to see three things: 1) an address where it can store the addresses of the Remember structures it creates; 2) the size of the requested memory block; and 3) flags indicating the type of memory being requested (CHIP, FAST, etc.).

The first item, the address of an address, might cause some problems. Intuition calls this magic address a "key." (We'll run across these again when we get to Animation Objects.) In Forth, we can set up this key using old-fashioned VARIABLEs. Declare a variable like this:

```
VARIABLE mykey 0 mykey !
```

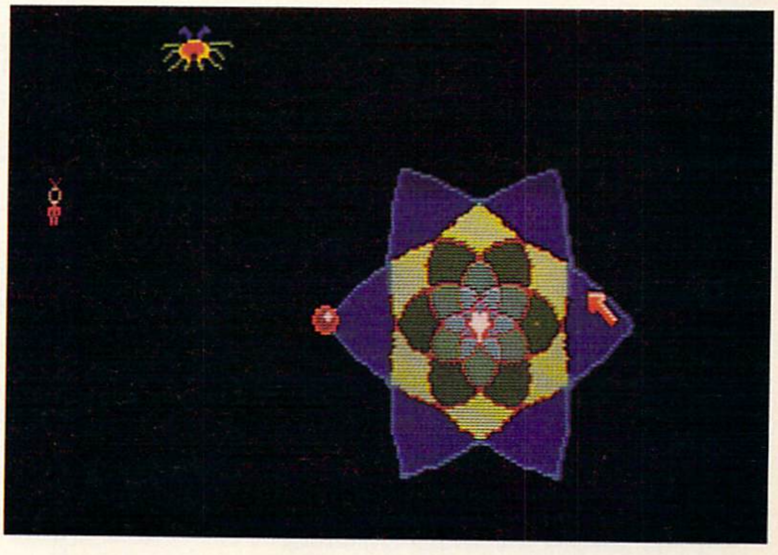
Later, when we need to pass the key to AllocRemember, we just say the name of our variable and its address is pushed onto the stack. For instance, we might call AllocRemember like so:

```
mykey size MEMF_CHIP AllocRemember to memptr
```

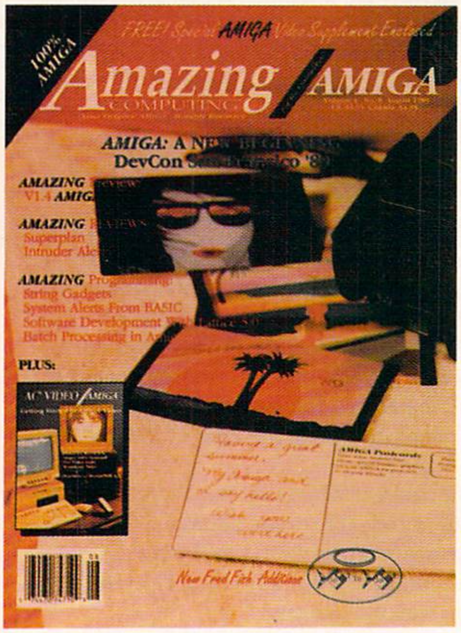
This statement calls the function and stores the result in the local or global variable called "memptr." If, after calling this function, we were to type

```
mykey @  <ENTER>
```

Putting it all together.



(continued page 97)



August 1989 V 4.8

Missing something special?

Don't miss another AC, Subscribe today!

For back issues, see page CIII.

Special! AMAZING Video Supplement

Getting Started in Video by Richard Starr
Part I of everyman's intro to great Amiga desktop video!
Magni 4004 Genlock by Oran Sands III
Amiga hardware for video professionals.
Pro Video Gold by Oran Sands III
An Amiga program for video pros.
MediaPhile by Larry Krieff
An affordable Amiga video editing system.
Broadcast Titler by R. Shamms Mortier, PhD.
Broadcast-quality titling on the Amiga.

Amazing Reviews

Superplan by Marion Deland
Grafox's new spreadsheet and time manager.
Intruder Alert by Mike Morrison
A new kind of Amiga monitor.
On Your Alert:
Using System Alerts from BASIC by John F. Wiederhirn
Display the Guru within BASIC & create your own messages.
Snapshot reviews by R. Brad Andrews
Hostage Rescue Mission, Combat Course & Thunder Blade.
International Soccer reviewed by Derek J. Perry
A weathery look at the game in the Amiga atmosphere.

Amazing Features

V1.4: A Pre Preview by Mike Morrison
A pre-release look at Amiga OS 1.4.
DevCon '89 by Mike Morrison
DevCon '89 in San Francisco, Mike tells all!
Hardcopy by Melissa J. Bernier
More puzzles to ponder for the Amiga enthusiast.

Amazing Programming

Lattice 5.0 by Gerald Hull
Experiences with the Lattice C Development System.
Executing Batch Files in AmigaBASIC by Mark Aydellotte
Executing AmigaDOS commands within AmigaBASIC programs.
Building a Better String Gadget by John Bushakra
Smooth data entry using string gadgets.

Amazing Columns

New Products and Other Neat Stuff by Elizabeth G. Fedorzyn
Elan Performer, IMG Scan, Dual Serial Boards, Archipelogos, Momentum Mail, TeleTutor and even The Three Bears!
Bug Bytes by John Steiner
A new Fat Agnus chip, Digi-Paint upgrade & more!
C Notes by Stephen Kemp
Directing programs via the Command Line.
Roomers by the Bandito
Whispers from Epyx, Atari, and the approaching hoard of European software products.

Index of Amazing Advertisers

Discover something interesting?
Need more information?

Use the Reader Service Card in every issue of AC to contact those advertisers who have sparked your interest. Advertisers want to hear from you. This is the best way they have of determining the Amiga community's interests and needs. Take a moment and contact the companies with products you want to know more about. And, if you wish to contact an Amazing Advertiser directly, please tell them you saw their advertisement in:

Amazing COMPUTING AMIGA
Your Original AMIGA Monthly Resource
For The Commodore

Advertiser	Page	Reader Service Number
ACDA	32	104
AmiEXPO	21	159
Anivision	31	150
AROCK Computer Software	52	107
Arrakis	54	108
ASDG	70	160
B & B Computers	69	110
Celestial Systems	60	112
Computability	66	115
Consultron	68	155
Creative Solutions	38	144
Delphi Noetic Systems	93	116
Empire Graphics	43	151
Expansion Technologies	97	120
Express-Way Software	53	122
EZ-Soft	4	119
Great Valley Products	1	158
Micro Momentum, Inc.	14	125
Micro-Systems Software	7	126
Micro-Systems Software	CII	101
MicroBotics	11	109
MicroWay	9	114
NewTek	CIV	102
Omnitech Computers	49	128
One Byte	87	129
Poor Person Software	40	131
Prespect Technics Inc.	17	133
Safe Harbor Software	103	134
Sedona Software	10	135
Software Advantage Consulting Corp.	101	138
Tangent 270	30	154
Templicity	46	157
Tensor Productions	38	141
The Bit Bucket Computer Store	51	142
The Memory Location	65	143

the address of "mykey" would be put onto the stack, and the "@" symbol would fetch 32 bits at that address. As a result, the number left on the stack is an address itself—or, more specifically, the address of an address. *Voilà!* What exactly is that magic address stored at "mykey"? It's a pointer to the first Remember structure in a linked list of Remember structures. Each node in the list keeps details on the size of the allocated memory block, the actual pointer to the memory, and, of course, a pointer to the next node.

In C, we would declare a variable and tell the compiler something about its type. In this case, the variable is a pointer to a Remember structure:

```
struct Remember *mykey = NULL;
```

When we actually call AllocRemember, we use the ampersand (&) to give us the address of this variable (the address of a variable that holds an address):

```
memptr = AllocRemember(&mykey,size,MEMF_CHIP);
```

We need to make three more allocations before we can use the Gel system. The "lastColor" array holds pointers to Virtual Sprite color tables. There are eight cells in the array, one for each hardware sprite. Here the system stores the address of the color table last used to display the respective sprite.

The "nextLine" array again has space for eight numbers. In this array, the system stores the next display line at which it may re-use the corresponding hardware sprite.

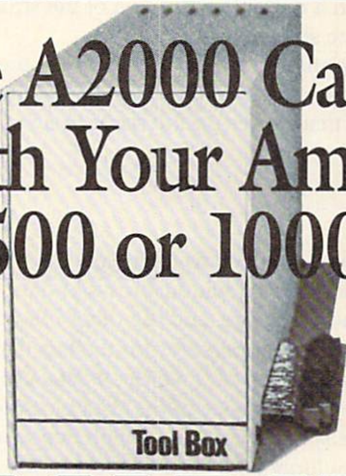
The last table we need to allocate is for collision handling. A "collTable" is an array of 16 pointers. Each pointer is the address of a routine executed when the system detects a certain kind of Gel-to-Gel collision. Using pointers to functions in Forth is difficult, because the routines must be written in machine language. The reason for this is that the Gel system executes these routines as machine language subroutines, and the function must end with an RTS instruction. Normal Forth definitions return through addresses left on the "return" stack. The only thing left to do is store the address of our GelsInfo structure in the RastPort and call InitGels. At this point, the Gel system has finally been initialized and is ready to use.

The first—and by far the simplest—in our array of functions is called "makehardsprt." As its name implies, it creates a hardware sprite—given the name of a disk file, a Remember key, and an array to store the hardware sprite pointer.

Since these functions will be using the AmigaDOS file I/O routines, you might want to include the Forth file, "libraries/dos.f," which contains the various file-access flag definitions. In my system, I've included both this file and a file defining Forth AmigaDOS calls. I have also made a snapshot, so that these definitions are ready to use when Multi-Forth loads. This file, "dos.calls," has been listed in the previous Gels articles, but for completeness, I've included it again as Listing Four.

The function "makehardsprt" opens the file and then reads the number of hardware sprites it contains. It then enters a loop, in which it reads each of the elements in the file. You might want to refer back to the file format for hardware sprites, which I described above.

The first block in the file is a SimpleSprite structure. The function allocates space for this in memory and reads it in from the file. Remember that in Multi-Forth, when you say the name



Use A2000 Cards With Your Amiga 500 or 1000.

TOOLBOX

EXPANSION SYSTEM

Don't get left behind in the ever growing market of expansion cards for the Amiga 2000. With the Toolbox 2-slot card cage you can use A2000 cards in your Amiga 500 or 1000.

GUARANTEED COMPATABILITY

Since we're the manufacturer we **guarantee** compatability with your machine. If there are any questions you make one phone call...the buck stops with us.

SAVE ON COMPLETE SYSTEMS

Since you buy **direct** from the manufacturer you save money. And complete systems save you even more.

A SELECTION OF PRODUCTS

Toolbox – 2-slot card cage with a 9amp power supply and bus pass-through.

Memory Card – A2000 type card with 2Mb of installed RAM, expandable to 8 megabytes.

Hard Card – Autobooting SCSI DMA A2000 controller in 32/40/48/60/80Mb sizes and more.

Systems	A	B	C	D	E
ToolBox	259.95	259.95	259.95	259.95	259.95
2Mb Memory	699.95			699.95	699.95
32Mb HardCard		749.95		749.95	
48Mb HardCard			799.95		799.95
Total Cost	959.90	1009.90	1059.90	1709.85	1759.85
Your Price	869.95	929.95	979.95	1599.95	1639.95

Call Today: 415/656-2890

VISA MasterCard

EXPANSION TECHNOLOGIES

44862 Osgood Road, Fremont, California 94539

Circle 120 on Reader Service card.

of a structure in a definition, the size of the structure in bytes is pushed onto the stack.

From information already contained in the SimpleSprite structure when it is read in, the program determines the size of the data block that makes up the sprite image. The formula is as follows:

```
size = (height + 2) * 2 * 2
```

A value of 2 is added to the height because sprite data is actually padded with 2 additional rows of data. One row proceeds the actual image, and the other follows it. The first multiplication by 2 compensates for the fact that each line of the sprite image is made up of 2 words of data. The next multiplication by 2 gives us the size of the data block in bytes. For example, if the sprite were 9 lines high, we would add 2 to get 11, and then multiply by 2 for a product of 22. This result represents the number of 16-bit words that make up the sprite image. We multiply by 2 again to get 44, the number of bytes that make up the image. We then allocate a block of memory to hold the image data and read the data from the disk.

Next, we have to store the SimpleSprite in the user's array. The function looks at the number of Gels in the file. If the number is greater than one, then the address passed to this function is assumed to be an array of pointers. The array is indexed by the loop index, multiplied by four (a pointer is four bytes long). If there is only one Gel in the file, then the Gel structure is assumed to be the address of a variable, and the pointer to the Gel structure is stored at the address. The loop continues for each sprite stored in the file. When the loop terminates, the file is closed, and the function returns.

The function "makevsprt" is similar to its hardware sprite companion. The main exception is the additional memory allocation that must take place in a virtual sprite. Specifically, the virtual sprite's collision mask and borderline image—both of which are used to detect collisions—must be retained. The following formulas are used in determining the size of these two blocks:

```
CollMask_size  = (Height * Width) * 2
BorderLine_size = Width * 2
```

From these formulas, it is evident that the collision mask must be as big as one plane of the sprite itself, and the borderline area must be as big as the sprite is wide. Therefore, if the image is 2 data words wide (16 pixels), we need to ask for space to store 2 words of data. Both sizes are multiplied by 2 to give the size in bytes, rather than in words.

The collision mask is the logical "or" of all the bit planes that make up the sprite image. The borderline image is the horizontal logical "or" of the collision mask. Think of the image bit planes as pages of a book. To make the collision mask, all the pages are put under a magic iron that squishes them into a single page. To make the borderline image, this page is then folded like an accordion, and put back under the iron. The iron squishes the accordion so that it becomes a single, narrow strip of paper.

After these memory areas have been allocated, "makevsprt" calls the Gel system function InitMasks, which performs the squishing described above. Again, if there is more than one Gel in the data file, it is assumed that the user passed an array to the function. The current Gel is stored in the array—indexed by "i"—which tells us how many iterations the loop has done. If only one Gel is stored in the file, the function assumes the user passed the address of a variable at which it can store the address of this Virtual Sprite. After reading all Gels, the loop terminates, the file is closed, and the function returns.

The next step on the Gel function evolutionary ladder is called "makebob," and by now you should be getting the hang of how these things work. The number of Gels stored in the file is read in, and the function enters a loop. In the loop, space is allocated for a bob, as well as, for a Virtual Sprite. From numbers already stored in the Virtual Sprite structure (using the converter), the size of the bob's image data is calculated. The formula is

```
(Width * Height * Depth) * 2
```

The depth variable is, of course, the number of bit planes in the image; once again, the multiplication by two gives the size of the image data block in bytes. The bob, virtual sprite, and image data are then read from the data file. The pointer to the image data is stored in the Virtual Sprite, then the two Gel structures are linked together. The bob drawing flags—OVERLAY, SAVEBACK, and SAVEBOB—are all saved with the converter and need not be set again.

The function then performs the same allocation as "makevsprt" for the bob's collision mask and borderline image. In addition, the same pointer to the collision mask area is stored in the bob's ImageShadow field. These two data areas are the same size. If the SAVEBACK flag was set with the converter, then the function will allocate the necessary memory area for saving and restoring the background when the Bob moves.

If the user requests a double-buffered bob, the converter stores a non-zero value in the bob's DBuffer field. When "makebob" sees this non-zero value, it allocates a data structure called "DBufPacket" and stores its pointer in the bob's DBuffer field. Also, "makebob" will allocate another memory block exactly the same size as the original block used to store the bob's image data. You can see that double-buffered bobs have a voracious appetite for memory—use them cautiously.

A call to InitMasks creates the masks for collision detection. The same mechanism described for hardware and virtual sprites stores the resulting Bob. When all bobs have been read and initialized, the data file is closed and the function returns.

The real star of this show is the function called "makeaob." As advertised, it reads in an animation object stored on disk and does the dirty work of linking it together. It turns out that linking the animation object is not half as bad as deciphering the reams of variables and flags that control these beasts. I'll go into a little more detail on how to control animation objects when we examine Listing Two. For now, let's concentrate on reading in and hooking together an animation object.

An animation object can be made up of several different components called AnimComps. A classic example of this is the "RoboCity" demo that nearly every Amigaphile has seen. It features a lanky bean-pole robot strolling along a city street. The demo also features a buxom female being, of sorts, a hopping creature, and, of course, the (in)famous dog that endlessly terrorizes a hapless fire hydrant.

Let me explain how animation objects are put together, using the robot from RoboCity as an example. The robot might be made up of six different animation components: one head, one body, two arms, and two legs. The position of each component is specified relative to the position of the Animation Object itself ([AnX, AnY] in the animation object data structure). In each AnimComp structure there is a pointer to the next AnimComp in the list (NextComp), as well as a pointer to the previous AnimComp (PrevComp).

Each component may have several different views. The robot's head might have two views, while his arms and legs might have three views each: forward, middle, and back. Each different view of a component is itself an AnimComp and is linked into the list by pointers called NextSeq and PrevSeq. Think of the entire structure as a two dimensional array with the different views of each component in the columns and each different component (head, arms, legs) in the rows. Figure 1 illustrates this concept.

The array "gacarr" is used to hold AnimComps as they are read in from the disk. The rows and columns are indexed by local variables w and x, respectively. When animation objects are created with the converter, each IFF image converted is assumed to be the next view in a sequence. In order to signal the end of a view sequence, the RINGTRIGGER gadget is selected. The flag is stored in the AnimComp structure and written to the disk. When reading back components, "makeaob" looks for any component with the RINGTRIGGER flag set. When the function finds the flag, the row index, w, is incriminated by one, and the column index, x, is set to zero. The next AnimComp in the file is then assumed to be the initial view of the next component making up the animation object. Figure 2 illustrates the hypothetical contents of the two-dimensional array after all components in the file have been read in. Note that the components have not yet been properly linked together.

Linking the components is performed by the second do-loop in "makeaob." First, the column index is set back to zero. The function then enters a loop that executes once for each component making up the object.

Inside the do-loop is a while-loop that links up each view of the current component. To do this, we must look in the cell that follows the current view. If we find a zero there, we know the current view is the last one for this component, and we must start on the next row of the array. If the value at the next cell is non-zero, we store that value in the current view's NextSeq pointer. We then store the current view in the next cell's PrevSeq pointer, increment the column index, and perform the next iteration of the while-loop.

When the while-loop terminates, we need to set the NextSeq pointer of the last view to point back to the initial view—thus creating a circular list. We then set the NextComp pointer of the initial view (x = 0) to the AnimComp at the next row in the array. We will also store the current component in the PrevComp pointer of the next component in the array. The do-loop then continues with the next row of the array.

There are only three more things to do after the do-loop terminates. We need to store the AnimComp at row zero, column zero in the HeadComp pointer of the AnimOb structure (the first component in the array). Then we call GetGBuffers, which handles allocation of the CollisionMask, BorderLine and ImageShadow for each component that makes up the object. A final call to InitMasks finishes the job. The data file is then closed and the dynamically allocated animation object is left on the stack for the calling program. Figure 3 illustrates an animation object that has been completely linked together.

There is a limit to the number of components you can store in a data file. If you look at the declaration of the array "gacarr," you will see it has been dimensioned to 10 rows by 10 columns. This means that the animation object must not have more than 10 components, and each component can have a maximum of 10 different views. It's important to remember just how memory hungry Gels actually are: Each component takes up 130 bytes of RAM: 60 bytes for the Virtual Sprite structure, 32 bytes for the Bob structure, and 98 bytes for the AnimComp structure. Ten such components would total 1,300 bytes. If each component had 10 views, you would be looking at 13,000 bytes of memory, and that does *not* include image data, collision masks, image shadows, borderlines or the buffers needed for saving and restoring the background. In short, Gels can be very useful gadgets, but you must be frugal in using them. They have a voracious appetite for RAM!

Now let's take a look at the short demo program in Listing Two. The demo uses images drawn with *DPaint II* which were run through the converter program. The hardware sprite is a pinball that bounces around in a star pattern; the Virtual Sprite is a little stick figure that does jumping jacks; the Bob is a spirograph pattern; and the animation object is a Centipede-like spider controlled with the joystick.

At the top of the program you will find all the variables needed to pass to the Gel functions. Two of the less obvious constants are "stickchip" and "sticktrig," which are actually hardware addresses. While shunned in the Amiga's early days, reading and writing directly to the hardware is now all the rage among programmers. (Ever the fashionable computer geek, I naturally follow this trend.)

The array "myvs" will hold the two virtual sprites. Each cell must be big enough to hold a four-byte address, so I ALLOT space for eight bytes. The remaining variables hold the pointers to each Gel created by the functions. The array "sprpath" holds the coordinate values of each point along the path followed by the hardware sprite.

The function "readcmap" reads in the colormap values from the disk. The file was created with the converter program and has the following format:

1.) Number of color registers (4 bytes).
2.) One word for each color register, stored in 0RGB form.

Space for the color table is allocated using AllocRemember, and a block of sized (numregs * 2) bytes are read from the file.

"Readstick" uses a quick and dirty method of detecting the direction in which the joystick is pointing and moves the animation object accordingly. We are interested in the 16 bits residing at memory location $DFF00C—specifically, bits 9, 8, 1, and 0. To inspect the state of these four bits at any moment in time, we must AND the contents of location $DFF00C with 771 (0000001100000011 binary = 771 decimal). Any of the four bits that are set in location $DFF00C will also be set as a result of this operation. The following table summarizes the possible results of this AND operation (bits are numbered zero to fifteen, zero being the bit on the far right):

```
0000000000000000 =   0   =   stick centered
0000000000000001 =   1   =   stick down
0000000000000010 =   2   =   stick down, right
0000000000000011 =   3   =   stick right
```

```
0000000100000000 =    256 =    stick up
0000000100000011 =    259 =    stick up, right
0000001000000000 =    512 =    stick up, left
0000001100000000 =    768 =    stick left
0000001100000001 =    769 =    stick down, left
```

The spider is controlled by storing appropriate values for its velocity. For example, if the stick is centered, the spider is not moving and has a velocity of zero. If the stick is pointing to the right, the spider will have a velocity of +3, which moves it to the right by 3 pixels. Recall from part one of this series that all values pertaining to animation objects—pixel values, velocities and accelerations—must be specified in fixed-point binary fraction notation. The decimal point is fixed in the word at bit location 6. This means that all whole-number values must be shifted 6 places to the left before they are stored. The Multi-Forth word "scale" does this for us.

Opening up the screen is done by the word "demoscreen." The screen is opened, the colormap is read and loaded (LoadRGB4), and the Gel system is initialized using "readygels."

The hardware sprite is read in by the word "initss." The call to "makehardsprt" has the following form:

```
0" df1:pinball.sprite" mykey myss makehardsprt
```

0" df1:pinball.sprite" puts the address of the NULL-terminated string onto the stack. The variables "mykey" and "myss" are used for Intuition's memory and storage of the pointer to a SimpleSprite structure.

Like all other resources the Amiga uses, hardware sprites must be requested and allocated. The call to GetSprite does this. A -1 passed to GetSprite indicates that any old sprite will do. If we wanted to be very specific, we could pass a value of 7 for instance, and GetSprite would try to give us sprite number 7. If it failed, it would return to us a -1.

To get the sprite ready for display, we call ChangeSprite, which expects as its parameters a ViewPort, a SimpleSprite, and a pointer to the sprite's image data. Notice that we had to fetch the address of our SimpleSprite with

```
myss @
```

since "myss" is a variable.

Initializing the virtual sprite is similar. The sprite's color table is stored in SprColors. There are two sprites stored in the file, so the same color table is stored in the second sprite as well. We can get at the second sprite with

```
myvs 4+ @
```

We must do 4+ before fetching the address of the virtual sprite from "myvs."; otherwise, we'll be storing a pointer in the wrong place. Before we can use the Virtual Sprite, it must be added to the system's Gel list, with AddVSprite. In order to keep things simple, I add only the first virtual sprite. To change the sprite's image I swap the image data pointers back and forth between the two sprites.

"Initbob" reads the spirograph pattern. Initial x- and y-coordinate values were stored with the converter program but did not work well with this demo, so I simply change these values after the Bob has been created. Like Virtual Sprites, Bobs must be added to the Gel list. Before drawing the Bob, the list must be sorted with SortGList. The pattern is then drawn and removed from the list, since it is only a backdrop over which the other Gels move.

The spider is created with "initaob." The pointer to the resulting AnimOb is stored in the global variable "aob." Again, initial values of XVel, YVel, RingXTrans and RingYTrans were stored with the converter but proved unsightly in this demo. The values of XVel and YVel are set to zero so that the spider does not move when it first appears. RingXTrans and Ring-YTrans are used to move the animation object when the system detects a component with the RINGTRIGGER flag set. When the flag is found, the system uses the NextSeq pointer to point back to the initial view in the sequence, and the values at RingX and RingY are added to the object's AnX and AnY values. I didn't want the spider to move when the RINGTRIGGER flag was found, so I had to store zeroes in RingX and RingY.

To add the animation object to the list in exactly the same fashion as AllocRemember a key is used. The contents of the variable "aobkey" are a pointer to the first AnimOb in a linked list of AnimObs.

Now the fun begins. The word "demo" uses the initialization routines above to build the Gel demo. All Gels are created, added to the list, and displayed. The program then enters a loop, which continues to execute until the fire button on the joystick is pressed.

First, the hardware sprite is moved. When the program detects the zero at the end of the path array, it resets the local variable "path" to point back to the beginning of the array. Otherwise, the sprite is moved to the current pixel values. To move the hardware sprite we must use the Intuition routines MakeScreen and RethinkDisplay. These two routines must also be used to display and move Virtual Sprites.

Next, the joystick is read and the program calls the Animate function, which moves the spider to its new location. Note that the spider is not actually drawn at this time. Animate processes every object in the list, so even if we had five animation objects, we would only need one call to Animate. The animation system uses a simple mechanism to switch between views of an object.

When you create an animation object, you store a non-zero value in the TimeSet field of the AnimOb data structure. This value is copied into another field called Timer. On each call to Animate, the value in Timer is incremented by negative one. When the value reaches zero, the system uses the NextSeq and PrevSeq pointers to switch to the next view of the component. The next view might have a TimeSet value different from that of the previous view, but this is perfectly legal. Remember that even these TimeSet values must be "scaled" or shifted six bit-positions to the left.

Before the Gels can be drawn, the list needs to be sorted. SortGlist does the sorting for us. Finally, one call to DrawGList draws the entire list of Gels.

The hardware sprite is then moved a second time, and the Virtual Sprite's image is toggled. Why bother to have two Virtual Sprites around if only one will be added to the system list? In this program, both sprites are the same size, so the question is quite valid. However, what if this weren't the case? In the real world, it is very likely that different views of an object will be different sizes. Even a difference of one or two pixels could cause problems, because the image data will not be interpreted correctly.

In all probability, if you were using these functions in a real graphics application, you would have one virtual sprite (or bob) that is actually added to the list and the array that is passed to the appropriate "Make" Gel function. When it came time to display a different view of the object, you would have to copy the width, height and depth values—as well as the image data pointer—into the virtual sprite data structure that you actually added to the list.

To check for the joystick trigger, we examine eight bits at location "sticktrig" ($BFE001). When the value at this location is 124, the trigger has been pressed, and the loop terminates.

Before exiting, we must remember to return to the system all those resources the demo program was using. The hardware sprite is returned with FreeSprite, the Virtual Sprite is removed from the list, and the screen is closed. Also, we mustn't forget the memory that is tied up in the spider's CollisionMask, BorderLine, and ImageShadow. The function FreeGBuffers frees up this memory. A zero is passed to FreeGBuffers to indicate that the spider was not double-buffered.

Finally, we free up all remaining memory with FreeRemember. The Remember key is passed, along with a non-zero value, to indicate that we want all memory freed. If we pass a value of zero to FreeRemember, the function deallocates only the space for the link nodes (the Remember structures) and leave the allocated memory blocks intact.

Well, this concludes the Gels in Multi-Forth series. I hope I was able to pass along some knowledge in an enjoyable, comprehensible way. If you are interested in using the updated converter program, look for it soon on an Amicus disk!

Even Up The Score!

INVESTOR'S ADVANTAGE

Let your Amiga give you the Advantage in making *better investment decisions!*

Color graphics of Individual Stocks and General Market Trends help you make more profit in this volatile market. High Low Close, Moving Averages, Centered Moving Averages, Volume, Relative Strength, Stochastics, Wilder's RSI, Cycles, Trend lines and Momentum. Powerful reports such as the Relative Strength Report help you pick the best performers. Use the Market Barometers to help you time your market entries. Update Stocks, Mutual Funds and Commodities manually or automatically. Easy to use communications included.

Only **$99.95**

See your local Dealer or Call:
Software Advantage Consulting Corporation
37346 Charter Oaks Blvd
Mt. Clemens, MI 48043 (313) 463-4995

Amiga and the Investor's Advantage are trademarks of their respective companies.

Circle 138 on Reader Service card.

Listing One: gels.f

```
\ All programs were written using Multi-Forth, from
\              Creative Solutions
\              4701 Randolph Rd. suite 12
\              Rockville MD  20852

\ ————————— Globals for Gels.f —————————
global fileid  0 to fileid  \ File handle returned by dosOpen
10 10 4 2array gacarr       \ Will hold AnimComps...
global idata   0 to idata   \ Address of ImageData read from
data files
global size    0 to size    \ Size in bytes of the ImageData.
global numgels 0 to numgels \ Number of gels in the data file.


( readygels initializes the Gel system )
( rport = address of a RastPort structure )
( key   = address of a Remember key variable )
: readygels ( rport / key — )
            0 0 0 locals| head tail ginfo rkey rport |

         rkey VirtualSprite MEMF_CLEAR AllocRemember to head
         rkey VirtualSprite MEMF_CLEAR AllocRemember to tail
         rkey GelsInfo      MEMF_CLEAR AllocRemember to
ginfo

          252 ginfo +gisprRsrvd c!  \ Reserve sprites 0 and
1.
          0 ginfo +gileftmost w!
          0 ginfo +gitopmost  w!

          rport +rpBitMap @ +bmBytesPerRow w@ 8* 1-
          ginfo +girightmost  w!

          rport +rpBitMap @ +bmRows w@ 1-
          ginfo +gibottommost w!

          rkey 4 8*  MEMF_CHIP MEMF_CLEAR | AllocRemember
          ginfo +gilastColor !

          rkey 2 8* MEMF_CHIP MEMF_CLEAR | AllocRemember
          ginfo +ginextLine  !

          rkey collTable MEMF_CHIP MEMF_CLEAR |
AllocRemember
          ginfo +gicollHandler !

          ginfo rport +rpGelsInfo !
          head tail ginfo InitGels ;

( ——————————————————————————— )
( makehardsprt — reads in hardware sprites from disk
)
( name  = address of a NULL terminated filename string.
)
( key   = address of a variable for Remembering.
)
( ssarr = either an address of a variable, or an array big
enough to )
(        hold all sprites in the file.
)
: makehardsprt  ( name / key / ssarr — )
  0 locals| ssprt ssarr rkey name |

  name MODE_OLDFILE dosOpen to fileid

  fileid addr.of numgels 4 dosRead drop

  numgels 0 do
```

```
              rkey SimpleSprite MEMF_CLEAR AllocRemember to
ssprt
              fileid ssprt SimpleSprite dosRead drop
              ssprt +ssheight w@ 2+  4* to size
              rkey size MEMF_CHIP MEMF_CLEAR | AllocRemember to
idata
              fileid idata size dosRead drop
              idata ssprt +ssposctldata !

              numgels 1 > if
                               ssprt ssarr i 4* + !
                          else
                               ssprt ssarr !
                          then
            loop

  fileid dosClose ;

( ------------------------------------------ )
( makevsprt - reads in virtual sprites from disk.
)
( name  =  address of a NULL terminated filename.
)
( key   =  address of a variable for Remembering.
)
( vsarr =  either an address or an array to store vsprites at.
)
: makevsprt  ( name / key / vsarr - )
  0 locals| lvsprt vsarr rkey name |

  name MODE_OLDFILE dosOpen to fileid

  fileid addr.of numgels 4 dosRead drop

  numgels 0 do
              rkey VirtualSprite MEMF_CLEAR AllocRemember to
lvsprt
              fileid lvsprt VirtualSprite dosRead drop
              lvsprt +vsHeight w@
              lvsprt +vsWidth  w@ *
              lvsprt +vsDepth  w@ * 2*  to size
              rkey size MEMF_CHIP  MEMF_CLEAR | AllocRemember
to idata
              fileid idata size dosRead drop
              idata lvsprt +vsImageData !

              rkey
              lvsprt +vsHeight w@ lvsprt +vsWidth w@ * 2*
              MEMF_CHIP MEMF_CLEAR | AllocRemember
              lvsprt +vsCollMask !

              rkey
              lvsprt +vsWidth w@ 2*
              MEMF_CHIP MEMF_CLEAR | AllocRemember
              lvsprt +vsBorderLine !

              lvsprt InitMasks

              numgels 1 > if
                             lvsprt vsarr i 4* + !
                          else
                             lvsprt vsarr !
                          then
            loop

   fileid dosClose ;

( ------------------------------------------ )
( makebob - read bobs in from disk...
)
( name  = NULL terminated filename
)
( key   = address of a variable for remembering.
)
( barr  = address of a variable or an array for bobs.......
)
( vsarr =    "    "  "   "    "  "   "   "  vsprites
)
: makebob  ( name / key / barr / vsarr - )
  0 0 locals| lbob vs vsarr barr rkey name |

  name MODE_OLDFILE dosOpen to fileid

  fileid addr.of numgels 4 dosRead drop

  numgels 0 do
              rkey VirtualSprite MEMF_CLEAR AllocRemember to vs
              rkey Bob           MEMF_CLEAR AllocRemember to
lbob
              fileid vs VirtualSprite dosRead drop
              fileid lbob Bob         dosRead drop
              vs +vsHeight w@
              vs +vsWidth  w@ *
              vs +vsDepth  w@ * 2* to size
              rkey size MEMF_CHIP MEMF_CLEAR | AllocRemember to
idata
              fileid idata size dosRead drop

              idata vs +vsImageData    !
              lbob  vs +vsVSBob        !
              vs lbob  +bobBobVSprite  !

              rkey
              vs +vsHeight w@ vs +vsWidth w@ * 2*
              MEMF_CLEAR MEMF_CHIP | AllocRemember dup
              vs +vsCollMask            !
              lbob  +bobImageShadow !

              rkey
              vs +vsWidth w@ 2*
              MEMF_CHIP MEMF_CLEAR | AllocRemember vs
+vsBorderLine !

              vs +vsFlags w@ SAVEBACK and
              if
                 rkey size MEMF_CLEAR MEMF_CHIP | AllocRemember
                 lbob +bobSaveBuffer !
              then

              bob +bobDBuffer @
              if
                 rkey DBufPacket MEMF_CLEAR AllocRemember
                 lbob +bobDBuffer !
                 rkey size MEMF_CLEAR MEMF_CHIP | AllocRemember
                 lbob +bobDBuffer @ +dbpbufbuffer !
              then

              vs InitMasks
              numgels 1 > if
                             vs vsarr i 4* + !
                             lbob  barr  i 4* + !
                          else
                             vs vsarr !
                             lbob  barr  !
                          then
            loop

  fileid dosClose ;

( ------------------------------------------ )
( makeaob - read in and link together an animation object.
)
( name   =  NULL terminated filename.
)
( key    =  address of a variable for Remembering.
)
( rport  =  address of a RastPort structure.
)
( db     =  TRUE (-1) or FALSE (0) flag indicating whether or
not      )
(           the AnimOb is double buffered.
)
: makeaob  ( name / key / rport / db - aob )
  0 0 0 0 0 0
  locals| lac lvs lbob laob w x db rport key name |
  ( w = row, x = col )

  10 0 do
```

```
         10 0 do
                0 j i gacarr !
              loop
       loop

  name MODE_OLDFILE dosOpen to fileid

  key AnimOb MEMF_CLEAR AllocRemember to laob

  fileid addr.of numgels 4 dosRead drop
  fileid laob AnimOb       dosRead drop

  numgels 0 do
              key VirtualSprite MEMF_CLEAR AllocRemember to lvs
              key Bob             MEMF_CLEAR AllocRemember to
lbob
              key AnimComp      MEMF_CLEAR AllocRemember to lac

              fileid lvs  VirtualSprite dosRead drop
              fileid lbob Bob           dosRead drop
              fileid lac  AnimComp      dosRead drop

              lvs +vsWidth w@ lvs +vsHeight w@ *
              lvs +vsDepth w@ * 2* to size
              key size MEMF_CLEAR MEMF_CHIP | AllocRemember to
idata

              fileid idata size dosRead drop

              idata lvs  +vsImageData   !
              lbob  lvs  +vsVSBob       !
              lvs   lbob +bobbobVSprite !
              lac   lbob +bobbobComp    !
              lbob  lac  +acAnimBob     !
              laob  lac  +acHeadOb      !

              lac  w x gacarr  !
              x 1+ to x

              lac +acFlags w@ RINGTRIGGER and
              if
                 w 1+ to w
                 0 to x
              then

            loop

  0 to x
  w 0 do
         begin
               i x 1+ gacarr @
         while
               i x 1+ gacarr @    ( -> | -> | -> | -> 0 )
               i x    gacarr @ +acNextSeq !

               i x    gacarr @    ( 0 <- | <- | <- | <- )
               i x 1+ gacarr +acPrevSeq   !
               x 1+ to x
         repeat

         i 0 gacarr @               ( Point around the circle )
         i x gacarr @ +acNextSeq !

         0 to x
         i 1+ x gacarr @
         i    x gacarr @ +acNextComp !

         i 1+ x gacarr @
         if
            i    x gacarr @
            i 1+ x gacarr @ +acPrevComp !
         then
      loop

  0 0 gacarr @ laob +aoHeadComp !
  laob rport db GetGBuffers
  laob InitGMasks
  drop
```

SAFE HARBOR
SOFTWARE & PERIPHERALS

Great Prices! Shipping based on weight and zone.
For Information & Catalog Call
Voice 414-544-2066
Pursuitable BBS 414-544-6567

Software	Price
3D Options	34.00
Ami Alignment Kit	36.00
ANIMagic	59.99
ARexx	31.10
B.A.D.	33.00
Baud Bandit	33.99
Blood Money	27.99
Caligari Consumer	165.00
Copyist	159.90
Cygnus Ed Prof	64.69
Digi Paint 3	61.99
Digi-View Gold	140.00
Elite	25.99
Jack Nicklaus Golf	32.99
Jet Master	61.99
K.C.S.	145.00
Kristal	32.99
Lattice C 5.0	190.00
Music X	195.00
Organize!	47.90
Pen Pal	90.00
PIXMate	42.30
Populous	36.99
Pro Video Gold	185.00
Video Page	87.09
TaleSpin	33.99
Superbase Prof 3.0	200.00
WShell	31.10

Hardware	Price
501 Clone, Supra	125.00
8-Up Board 0K	179.00
AMAX	135.00
Fast Card 2000, Xetec	170.00
flickerFixer	475.00
Floppy Drive (Int)	115.00
Floppy Drive, Unidrive	140.00
Han-D-Scan	320.00
Harddrive, ST296N	550.00
HardFrame 2000	250.00
IMG Scan	120.00
Joystick, Gravis	39.99
Midget Racer	350.00
Modem, Supra	135.00
MIDI, ECE	50.00
SCSI Con., Kronos	220.00
SCSI, Supra 500/2000	185.00
Spirit Board 0K	215.00
Spirit Board, SIN 0K	215.00

Orders Only Please:
Outside WI: **800-544-6599**
Inside WI: **414-544-6599**
Visa/MC/CODs

2414 Pendleton Place ■ Waukesha. WI 53188 ■ 9 AM to 5 PM M-F

Circle 134 on Reader Service card.

```
  fileid dosClose
  laob ;
```

Listing Two: demo.f

```
global cmap     ( Holds address of dynamically allocated color
map. )
global sc       ( For easy access to the address of my screen.
)
global hardsprite  ( The number of the hardware sprite returned
by )
                   ( GetSprite.
)
global aob      ( Address of the AnimOb returned by makeaob.
)
hex
   dff00c constant stickchip  ( Joystick hardware address.
)
   bfe001 constant sticktrig  ( Joystick fire button.
)
decimal
create   myvs    8 allot
         myvs 8 erase         ( Holds two pointers to vsprites
)
variable myss    0 myss    !  ( A spot to store the
SimpleSprite )
variable mykey   0 mykey   !  ( Variable for Remembering......
)
variable aobkey  0 aobkey  !  ( Variable for AnimOb key
)
variable mybob   0 mybob   !  ( Will hold the address of a bob.
)
variable mybobvs 0 mybobvs !  ( Holds the bob's vsprite struct.
)
struct NewScreen ns           ( The demo
screen................. )
ns initscreen                    ( Initialize the screen.
)
        •
hex
create sprcols                   ( The virtual sprite's color
table )
        0 w, 0eca w, 0f00 w, 0ff0 w,
decimal
                                 ( Path for the hardware sprite
create sprpath
160 c, 19 c,  159 c, 22 c,  158 c, 25 c,  157 c, 28 c,  156 c,
31 c,
155 c, 34 c,  154 c, 37 c,  153 c, 40 c,  152 c, 43 c,  151 c,
46 c,
150 c, 49 c,  149 c, 52 c,  148 c, 55 c,  147 c, 58 c,  146 c,
61 c,
145 c, 64 c,  144 c, 67 c,  143 c, 70 c,  142 c, 73 c,  141 c,
76 c,
140 c, 79 c,  139 c, 82 c,  138 c, 85 c,  137 c, 88 c,  136 c,
91 c,
135 c, 94 c,  134 c, 98 c,  133 c, 101 c, 132 c, 104 c, 131 c,
107 c,
130 c, 110 c, 129 c, 113 c, 128 c, 116 c, 127 c, 119 c, 126 c,
122 c,
125 c, 125 c, 124 c, 128 c, 123 c, 131 c, 122 c, 134 c, 121 c,
137 c,
120 c, 140 c, 119 c, 143 c, 118 c, 146 c, 117 c, 149 c, 116 c,
152 c,
115 c, 155 c, 114 c, 158 c, 113 c, 161 c, 112 c, 164 c,

112 c, 165 c, 114 c, 164 c, 116 c, 163 c,
118 c, 161 c, 120 c, 160 c,
122 c, 158 c, 124 c, 157 c, 126 c, 155 c,
128 c, 154 c, 130 c, 153 c,
132 c, 151 c, 134 c, 150 c, 136 c, 148 c,
138 c, 147 c, 140 c, 145 c,
142 c, 144 c, 144 c, 142 c, 146 c, 141 c,
148 c, 140 c, 150 c, 138 c,
152 c, 137 c, 154 c, 135 c, 156 c, 134 c,
158 c, 132 c, 160 c, 131 c,
162 c, 129 c, 164 c, 128 c, 166 c, 127 c,
168 c, 125 c, 170 c, 124 c,
172 c, 122 c, 174 c, 121 c, 176 c, 119 c,
178 c, 118 c, 180 c, 117 c,
182 c, 115 c, 184 c, 114 c, 186 c, 112 c,
188 c, 111 c, 190 c, 109 c,
192 c, 108 c, 194 c, 106 c, 196 c, 105 c,
198 c, 104 c, 200 c, 102 c,
202 c, 101 c, 204 c, 99 c,  206 c, 98 c,
208 c, 96  c, 210 c, 95 c,
212 c, 93  c, 214 c, 92 c,  216 c, 91 c,
218 c, 89  c, 220 c, 88 c,
222 c, 86  c, 224 c, 85 c,  226 c, 83 c,
228 c, 82  c, 230 c, 81 c,
232 c, 79  c, 234 c, 78 c,  236 c, 76 c,

237 c, 75 c, 235 c, 75 c, 233 c, 75 c,
231 c, 75 c, 229 c, 75 c, 227 c, 75 c,
225 c, 75 c, 223 c, 75 c,
221 c, 75 c, 219 c, 75 c, 217 c, 75 c,
215 c, 75 c, 213 c, 75 c,
211 c, 75 c, 209 c, 75 c, 207 c, 75 c,
205 c, 75 c, 203 c, 75 c,
201 c, 75 c, 199 c, 75 c, 197 c, 75 c,
195 c, 75 c, 193 c, 75 c,
191 c, 75 c, 189 c, 75 c, 187 c, 75 c,
185 c, 75 c, 183 c, 75 c,
181 c, 75 c, 179 c, 75 c, 177 c, 75 c,
175 c, 75 c, 173 c, 75 c,
171 c, 75 c, 169 c, 75 c, 167 c, 75 c,
165 c, 75 c, 163 c, 75 c,
161 c, 75 c, 159 c, 75 c, 157 c, 75 c,
155 c, 75 c, 153 c, 75 c,
151 c, 75 c, 149 c, 75 c, 147 c, 75 c,
145 c, 75 c, 143 c, 75 c,
141 c, 75 c, 139 c, 75 c, 137 c, 75 c,
135 c, 75 c, 133 c, 75 c,
131 c, 75 c, 129 c, 75 c, 127 c, 75 c,
125 c, 75 c, 123 c, 75 c,
121 c, 75 c, 119 c, 75 c, 117 c, 75 c,
115 c, 75 c, 113 c, 75 c,
111 c, 75 c, 109 c, 75 c, 107 c, 75 c,
105 c, 75 c, 103 c, 75 c,
101 c, 75 c, 99 c, 75 c,  97 c, 75 c,
95 c, 75 c,  93 c, 75 c,
91 c, 75 c,  89 c, 75 c,  87 c, 75 c,
85 c, 75 c,  83 c, 75 c,

84 c, 75 c, 86 c, 77 c, 88 c, 78 c, 90 c, 80 c, 92 c, 81 c,
94 c, 82 c, 96 c, 84 c, 98 c, 85 c, 100 c, 87 c, 102 c, 88 c,
104 c, 90 c, 106 c, 91 c, 108 c, 93 c, 110 c, 94 c, 112 c, 96
c,
114 c, 97 c, 116 c, 98 c, 118 c, 100 c, 120 c, 101 c, 122 c,
103 c,
124 c, 104 c, 126 c, 106 c, 128 c, 107 c, 130 c, 109 c, 132 c,
110 c,
134 c, 111 c, 136 c, 113 c, 138 c, 114 c, 140 c, 116 c, 142 c,
117 c,
144 c, 119 c, 146 c, 120 c, 148 c, 122 c, 150 c, 123 c, 152 c,
125 c,
154 c, 126 c, 156 c, 127 c, 158 c, 129 c, 160 c, 130 c, 162 c,
132 c,
164 c, 133 c, 166 c, 135 c, 168 c, 136 c, 170 c, 138 c, 172 c,
139 c,
174 c, 140 c, 176 c, 142 c, 178 c, 143 c, 180 c, 145 c, 182 c,
146 c,
184 c, 148 c, 186 c, 149 c, 188 c, 151 c, 190 c, 152 c, 192 c,
154 c,
194 c, 155 c, 196 c, 156 c, 198 c, 158 c, 200 c, 159 c, 202 c,
161 c,
204 c, 162 c, 206 c, 164 c,

207 c, 164 c, 206 c, 161 c, 205 c, 158 c, 204 c, 155 c, 203 c,
152 c,
202 c, 149 c, 201 c, 146 c, 200 c, 142 c, 199 c, 139 c, 198 c,
136 c,
197 c, 133 c, 196 c, 130 c, 195 c, 127 c, 194 c, 124 c, 193 c,
121 c,
192 c, 118 c, 191 c, 115 c, 190 c, 111 c, 189 c, 108 c, 188 c,
105 c,
187 c, 102 c, 186 c, 99 c, 185 c, 96 c, 184 c, 93 c, 183 c, 90
c,
182 c, 87 c, 181 c, 84 c, 180 c, 80 c, 179 c, 77 c, 178 c, 74
c,
177 c, 71 c, 176 c, 68 c, 175 c, 65 c, 174 c, 62 c, 173 c, 59
c,
172 c, 56 c, 171 c, 53 c, 170 c, 49 c, 169 c, 46 c, 168 c, 43
c,
167 c, 40 c, 166 c, 37 c, 165 c, 34 c, 164 c, 31 c, 163 c, 28
c,
162 c, 25 c, 161 c, 20 c,

0 c,

( ——————————————————————————— )
( readcmap – read in color map values from disk.
)
: readcmap
          0 0 locals| fileid numregs |

        0" df1:article3.cmap" MODE_OLDFILE dosOpen to fileid
        fileid addr.of numregs 4 dosRead drop
        mykey numregs 2* MEMF_CHIP MEMF_CLEAR | AllocRemember
to cmap


        fileid cmap numregs 2* dosRead drop
        fileid dosClose ;

( ——————————————————————————— )
( readstick – examine contents of $DFF00C, AND with 771 and
)
(               take appropriate action.
)
```

```
: readstick
              stickchip w@  771 and
              case
                   ( stick is centered. )
                   0  of  0 aob +aoxvel w!
                          0 aob +aoyvel w!
                      endof

                   ( stick is pulled back. )
                   1  of  0 aob +aoxvel w!
                          4 6 scale aob +aoyvel w!
                      endof

                   ( stick is back, right. )
                   2  of  3 6 scale aob +aoxvel w!
                          4 6 scale aob +aoyvel w!
                      endof

                   ( stick is right        )
                   3  of  3 6 scale aob +aoxvel w!
                          0 aob +aoyvel w!
                      endof

                   ( stick is up.          )
                 256  of  0 aob +aoxvel w!
                         -4 6 scale aob +aoyvel w!
                      endof

                   ( stick is up, right    )
                 259  of  3 6 scale aob +aoxvel w!
                         -4 6 scale aob +aoyvel w!
                      endof

                   ( stick is up, left     )
                 512  of  -3 6 scale aob +aoxvel w!
                          -4 6 scale aob +aoyvel w!
                      endof

                   ( stick is left         )
                 768  of  -3 6 scale aob +aoxvel w!
                           0 aob +aoyvel w!
                      endof

                   ( stick is down, left.  )
                 769  of  -3 6 scale aob +aoxvel w!
                           4 6 scale aob +aoyvel w!
                      endof
              endcase ;

( ---------------------------------- )
( demoscreen - open up the screen, load the color map,
initialize )
(              the gel system.
)
: demoscreen
              0 ns +nsviewmodes w!
              SPRITES ns +nsviewmodes w!
              ns +nsType dup w@ SCREENQUIET | swap w!
              ns openscreen
              currentscreen @ to sc
               readcmap
               sc +scViewPort cmap 32 LoadRGB4
               sc +scRastPort mykey readygels  ;

( ---------------------------------- )
( initss - read the pinball image from disk, allocate a
hardware )
(           sprite, and get the sprite ready for display.
)
: initss
              0" dfl:pinball.sprite" mykey myss makehardsprt
              myss @ -1 GetSprite to hardsprite
              sc +scViewPort myss @ dup +ssposctldata @
ChangeSprite ;

( ---------------------------------- )
( initvs - read in the stick figure, store pointers to color
map )
(           table.  Add the vsprite to the list.
)
: initvs
              0" dfl:stickfig.vsprite"  mykey myvs  makevsprt
              sprcols myvs @  +vssprcolors !
              sprcols myvs 4+ @ +vssprcolors !
              myvs @ sc +scRastPort AddVSprite ;

( ---------------------------------- )
( initbob - read the spirograph, change x & y values, add the
)
(           bob, draw it, and remove it, so it doesn't clutter
)
(           the list.
)
: initbob
              0" dfl:spiro.bob" mykey mybob mybobvs makebob
              110 mybobvs @ +vsx w!
               50 mybobvs @ +vsy w!
              mybob @ sc +scRastPort AddBob
              sc +scRastPort SortGList
              sc +scRastPort sc +scViewPort DrawGList
              mybob @ sc +scRastPort sc +scViewPort RemIBob ;

( ---------------------------------- )
( initaob - read in the spider, set its velocities and ring
)
(           translation values to zero, add the AnimOb.
)
: initaob
              0" dfl:spider.aob" mykey sc +scRastPort 0
makeaob to aob
              0  aob +aoxvel w!
              0  aob +aoyvel w!
              0  aob +aoRingXTrans w!
              0  aob +aoRingYTrans w!

              aob aobkey sc +scRastPort AddAnimOb ;

( --------------------------------- )
( demo - open the screen, initialize all Gels, loop until
)
(         joystick fire button is pressed.
)
: demo
        0 sprpath locals| path vstoggle |

        demoscreen

        initvs
        initbob
        initss
        initaob

        begin
           path c@ 0=
           if
             sprpath to path
           then

           sc +scViewPort myss @

           path c@  path 1+ c@  MoveSprite
           path 2+ to path

           sc MakeScreen
           RethinkDisplay

           readstick
           aobkey sc +scRastport Animate
           sc +scRastPort Sortglist
           sc +scRastPort sc +scViewPort Drawglist

           path c@ 0=
           if
             sprpath to path
           then
```

```
            sc +scViewPort myss @

            path c@  path 1+ c@  MoveSprite
            path 2+ to path

            sc MakeScreen
            RethinkDisplay

         vstoggle if
                    myvs @ +vsimagedata @
                    myvs 4+ @ +vsimagedata @ myvs @
+vsimagedata !
                    myvs 4+ @ +vsimagedata !
                    vstoggle not to vstoggle
                  else
                    vstoggle not to vstoggle
                  then
         sticktrig c@ 124 =
        until

        hardsprite FreeSprite
        myvs @ RemVSprite
        sc CloseScreen
        aob sc +scRastPort 0 FreeGBuffers
        mykey -1 FreeRemember ;

/*
```

Listing Three : gels.c

```
>>  Functions for initializing Gels...
*/

#include <exec/types.h>
#include <exec/memory.h>
#include <graphics/sprite.h>
#include <graphics/gels.h>
#include <graphics/rastport.h>
#include <intuition/intuition.h>
#include <libraries/dos.h>
#include <proto/intuition.h>
#include <proto/dos.h>
#include <proto/graphics.h>
/* ─────────────────────────── */
/*
>  ReadyGels prepares the system for use.
>  This is lifted pretty much intact from the Rom Kernel
>  Manual...Except all my allocation is done with
AllocRemember().
*/
void ReadyGels(rport,key)
struct RastPort *rport;
struct Remember **key;
{
     struct VSprite  *head  = NULL;
     struct VSprite  *tail  = NULL;
     struct GelsInfo *ginfo = NULL;

     head  = (struct VSprite *)
             AllocRemember(key,sizeof(struct
VSprite),MEMF_CLEAR);
     tail  = (struct VSprite *)
             AllocRemember(key,sizeof(struct
VSprite),MEMF_CLEAR);
     ginfo = (struct GelsInfo *)
             AllocRemember(key,sizeof(struct
GelsInfo),MEMF_CLEAR);

     ginfo -> sprRsrvd = -1;
     ginfo -> leftmost = 0;
     ginfo -> topmost  = 0;
     ginfo -> rightmost  = rport->BitMap->BytesPerRow * 8 - 1;
     ginfo -> bottommost = rport->BitMap->Rows - 1;

     ginfo -> lastColor = (WORD **)
                          AllocRemember(key,sizeof(LONG) * 8,
                                                     MEMF_CHIP |
MEMF_CLEAR);

     ginfo -> nextLine  = (WORD *)
                          AllocRemember(key,sizeof(WORD) * 8,
                                                     MEMF_CHIP |
MEMF_CLEAR);

     ginfo -> collHandler = (struct collTable *)
                            AllocRemember(key,sizeof(struct
collTable),
                                                       MEMF_CHIP |
MEMF_CLEAR);

     rport -> GelsInfo = ginfo;
     InitGels(head,tail,ginfo);
} /* ReadyGels */

/* ─────────────────────────── */
/*
>  MakeHardSprt(name,key,ssarr)
>  Synopsis:  Reads in data stored in the user's file, sets
>             up a hardware sprite.
>  Inputs:    name: NULL terminated filename.
>             key:  pointer to a Remember struct.
>             ssarr: either an array of SimpleSprite pointers,
>                    or if there is only one gel in the file,
>                    the address of a pointer to a
SimpleSprite.
>  Returns:   an initialized SimpleSprite, or an array of
>             SimpleSprites.
*/
void MakeHardSprt(name,key,ssarr)
char   *name;
struct Remember **key;
struct SimpleSprite **ssarr;
{

     BPTR fileid = NULL;
     struct SimpleSprite *ssprt = NULL;
     WORD *idata = NULL;
     int i;
     long numgels    = 0;  /* number of sprites in the file.
*/
     long size       = 0;  /* size of the image data, in bytes.
*/


     fileid = Open(name,MODE_OLDFILE);

     Read(fileid,(char *)&numgels,4L);

for(i = 0; i < numgels; i++)
   {
      ssprt = (struct SimpleSprite *)
              AllocRemember(key,sizeof(struct
SimpleSprite),MEMF_CLEAR);

      Read(fileid,(char *)ssprt,(long)sizeof(struct
SimpleSprite));
      size  = ((ssprt->height + 2L)) * 4L;
      idata = (WORD *)AllocRemember(key,size,MEMF_CHIP |
MEMF_CLEAR);
      Read(fileid,(char *)idata,(long)size);

      ssprt -> posctldata = idata;

  /* If there is more than one gel, we're storing in an array,
*/
  /* otherwise, we're storing in a pointer.
*/

      if(numgels > 1)
        ssarr[i] = ssprt;
      else
        *ssarr = ssprt;
```

```
    } /* for */

   Close(fileid);
} /* MakeHardSprt */

/* ---------------------------- */
/*
>  MakeVSprt(name,key,vsarr)
>  Synopsis: Reads in data stored in a VSprite file.
>  Inputs:   name: NULL terminated file name.
>            key:  pointer to a Remember struct.
>            vsarr: if more than one VS in the file, an array
of
>                   pointers to VSprites, otherwise, it's an
>                   address of a pointer to a VSprite.
>  Returns:  An initialized VSprite, or array of Vsprites.
*/
void MakeVSprt(name,key,vsarr)
char   *name;
struct Remember **key;
struct VSprite  **vsarr;
{

     BPTR fileid  = NULL;
     struct VSprite *vsprt = NULL;
     WORD *idata  = NULL;
     int i;
     long numgels    = 0;
     long size       = 0;


     fileid = Open(name,MODE_OLDFILE);

     Read(fileid,(char *)&numgels,4L);

for(i = 0; i < numgels; i++)
  {
   vsprt = (struct VSprite *)
          AllocRemember(key,sizeof(struct
VSprite),MEMF_CLEAR);

   Read(fileid,(char *)vsprt,(long)sizeof(struct VSprite));

   size  = (vsprt->Height * vsprt->Width * vsprt->Depth) * 2L;
   idata = (WORD *)AllocRemember(key,size,MEMF_CHIP |
MEMF_CLEAR);
   Read(fileid,(char *)idata,(long)size);

   vsprt -> CollMask = (WORD *)
                   AllocRemember(key,
                             (vsprt->Height * vsprt-
>Width) * 2L,
                              MEMF_CLEAR | MEMF_CHIP);

   vsprt -> BorderLine = (WORD *)
                    AllocRemember(key,vsprt->Width * 2L,
                              MEMF_CLEAR | MEMF_CHIP);

   vsprt -> ImageData = idata;

   InitMasks(vsprt);

   if(numgels > 1)
     vsarr[i] = vsprt;
   else
     *vsarr = vsprt;

  } /* for */

  Close(fileid);
} /* MakeVSprt */

/* ---------------------------- */
/* MakeBob(name,key,barr,vsarr)
>  Synopsis:  Reads in and initializes a bob, or an array of
>             bobs.
>  Inputs:    name: NULL terminated file name.
>             key:  pointer to a Remember struct.
>             barr: array of pointers to Bobs, or an address
>                   of a pointer to a Bob if only one gel in
>                   the file.
>             vsarr: same as above, only VSprites.
>  Returns:   initialized bob, or an array of bobs.
*/
void MakeBob(name,key,barr,vsarr)
char   *name;
struct Remember **key;
struct Bob     **barr;
struct VSprite **vsarr;
{
     struct Bob        *bob   = NULL;
     struct VSprite    *vs    = NULL;
     WORD              *idata = NULL;
     BPTR              fileid = NULL;
     int i;
     long  numgels = 0;
     long  size    = 0;

     fileid = Open(name,MODE_OLDFILE);
     Read(fileid,(char *)&numgels,4L);

 for(i = 0; i < numgels; i++)
    {
      vs  = (struct VSprite *)
            AllocRemember(key,sizeof(struct
VSprite),MEMF_CLEAR);
      bob = (struct Bob *)
            AllocRemember(key,sizeof(struct Bob),MEMF_CLEAR);

      Read(fileid,(char *)vs,(long)sizeof(struct VSprite));
      Read(fileid,(char *)bob,(long)sizeof(struct Bob));

      size = (vs->Height * vs->Width * vs->Depth) * 2L;
      idata = (WORD *)AllocRemember(key,size,MEMF_CLEAR |
MEMF_CHIP);
      Read(fileid,(char *)idata,size);

      vs  -> ImageData  = idata;
      vs  -> VSBob      = bob;
      bob -> BobVSprite = vs;

      vs->CollMask = bob->ImageShadow =
          (WORD *)AllocRemember(key,vs->Height * vs->Width *
2L,
                               MEMF_CLEAR | MEMF_CHIP);

      vs->BorderLine = (WORD *)AllocRemember(key,vs->Width *
2L,
                                     MEMF_CLEAR |
MEMF_CHIP);

      if(vs->Flags & SAVEBACK)
        bob->SaveBuffer = (WORD *)
                     AllocRemember(key,size,MEMF_CLEAR |
MEMF_CHIP);

      if(bob -> DBuffer != NULL)
        {
          bob -> DBuffer = (struct DBufPacket *)
                         AllocRemember(key,sizeof(struct
DBufPacket),
                                  MEMF_CLEAR);
          bob -> DBuffer->BufBuffer =
                      (WORD *)AllocRemember(key,size,
                                       MEMF_CLEAR |
MEMF_CHIP);
        }
      InitMasks(vs);
      if( numgels > 1 )
       {
        vsarr[i] = vs;
        barr[i] = bob;
       }
      else
```

```
       {
        *vsarr = vs;
        *barr = bob;
       }
   } /* for */
   Close(fileid);
} /* MakeBob */

/* ______________________________ */
/* MakeAOB(name,key,rport,db)
>  Synopsis:  Creates and links together an animation object
>             from data stored in a file.
>  Inputs:    name: NULL terminated file name.
>             key:  pointer to a Remember struct.
>             rport: pointer to a RastPort.
>             db:   an int, TRUE if user wants double
buffering,
>                   FALSE otherwise.
>  Returns:   returns a pointer to an AnimOb via a 'return'
>             statement.
*/
struct AnimOb *MakeAOB(name,key,rport,db)
char *name;
struct Remember **key;
struct RastPort *rport;
int db;
{
         struct AnimComp *acarr[10][10];
         struct AnimComp *ac  = NULL;
         struct VSprite  *vs  = NULL;
         struct Bob      *bob = NULL;
         struct AnimOb   *aob = NULL;
         WORD          *idata = NULL;
         BPTR          fileid = NULL;
         long numdone  = 0;
         long size     = 0;
         int i,k;                /* various loop control vars */
         int h = 0; int j = 0; /* j = row, h = column.....  */

 /* The 10x10 array of AnimComps needs to be cleared of
garbage. */
    for(i = 0; i < 10; i++)
      for(k = 0; k < 10; k++)
         acarr[i][k] = NULL;

    fileid = Open(name,MODE_OLDFILE);

    aob = (struct AnimOb *)
           AllocRemember(key,sizeof(struct AnimOb),MEMF_CLEAR);

    Read(fileid,(char *)&numdone,4L);
    Read(fileid,(char *)aob,sizeof(struct AnimOb));

    for(i = 0; i < numdone; i++)
      {
        vs  = (struct VSprite *)
             AllocRemember(key,sizeof(struct
VSprite),MEMF_CLEAR);
        bob = (struct Bob *)
              AllocRemember(key,sizeof(struct Bob),MEMF_CLEAR);
        ac  = (struct AnimComp *)
              AllocRemember(key,sizeof(struct
AnimComp),MEMF_CLEAR);

        Read(fileid,(char *)vs,sizeof(struct VSprite));
        Read(fileid,(char *)bob,sizeof(struct Bob));
        Read(fileid,(char *)ac,sizeof(struct AnimComp));

        size  = vs->Width * vs->Height * vs->Depth * 2L;
        idata = (WORD *)AllocRemember(key,size,MEMF_CLEAR |
MEMF_CHIP);

        Read(fileid,(char *)idata,size);

        vs  -> ImageData  = idata;
        vs  -> VSBob      = bob;
        bob -> BobVSprite = vs;
        bob -> BobComp    = ac;
        ac  -> AnimBob    = bob;
        ac  -> HeadOb     = aob;

        acarr[j][h] = ac;
        h++;
        if( ac -> Flags & RINGTRIGGER ) /* time to start */
         {                                 /* on a new row? */
           j++; h = 0;
         }

      } /* for */

   h = 0;
   j--;
   for(i = 0; i <= j; i++)
     {
        while(acarr[i][h+1] != NULL)
           {
              acarr[i][h] -> NextSeq = acarr[i][h + 1];
              acarr[i][h + 1] -> PrevSeq = acarr[i][h];
              h++;
           }
        acarr[i][h] -> NextSeq = acarr[i][0];
        h = 0;
        acarr[i][h]->NextComp = acarr[i + 1][h];
        if(acarr[i + 1][h] != NULL)
          acarr[i + 1][h] -> PrevComp = acarr[i][h];
     } /* for */

     aob -> HeadComp = acarr[0][0];
     GetGBuffers(aob,rport,db);
     InitGMasks(aob);

     Close(fileid);
     return(aob);
} /* MakeAOB */
```

Listing Four: dos.calls

```
\ ****  Dos function calls, as per Multi-Forth.
\ **    I've prefaced each function with 'dos' to avoid
\ **    confusion with Multi-Forth's file functions.

: dosOpen       !d2 !d1  call.lib 1 5  @d0 ;
: dosClose      !d1      call.lib 1 6 ;
: dosRead       !d3 !d2 !d1   call.lib 1 7 @d0 ;
: dosWrite      !d3 !d2 !d1   call.lib 1 8 @d0 ;
: dosIoErr      call.lib 1 22 @d0 ;
: dosDelete     !d1  call.lib 1 12 ;
: dosExamine    !d2 !d1 call.lib 1 17 @d0 ;
: dosExNext     !d2 !d1 call.lib 1 18 @d0 ;
: CurrentDir    !d1 call.lib 1 21  @d0 ;
: ParentDir     !d1 call.lib 1 35 @d0 ;
: dosLock       !d2  !d1 call.lib 1 14 @d0 ;
: dosUnlock     !d1 call.lib 1 15 ;
: dosExit       !d1 call.lib 1 24 ;
: dosLoadSeg    !d1 call.lib 1 25 @d0 ;
: dosUnLoadSeg !d1 call.lib 1 26 ;
: dosCreateProc   !d4 !d3 !d2 !d1 call.lib 1 23 @d0 ;
```

•AC•

Be an

Amazing COMPUTING AMIGA

Your Original AMIGA Monthly Resource

Writer!

Call or write Today for our Writer's Guide.

The AMICUS & Fred Fish Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

AMICUS Disk 1

ABasic programs: Graphics

3DSolids	3d solids modeling prog. w/sample data files
Blocks	draws blocks
Cubes	draws cubes
Durer	draws pictures in the style of Durer
FScape	draws fractal landscapes
Hidden	3D drawing program, w/ hidden line removal
JPad	simple paint program
Optical	draw several optical illusions
PaintBox	simple paint program
Shuttle	draws the Shuttle in 3d wireframe
SpaceArt	graphics demo
Speaker	speech utility
Sphere	draws spheres
Spiral	draws color spirals
ThreeDee	3d function plots
Topography	artificial topography
Wheels	draws circle graphics
Xenos	draws fractal planet landscapes

ABasic programs: Tools

AddressBook	simple database program for addresses
CardFile	simple card file database program
Demo	multiwindow demo
KeyCodes	shows keycodes for a key you press
Menu	run many ABasic programs from a menu
MoreColors	way to get more colors on the screen at once, using aliasing
shapes	simple color shape designer SpeakIt speech and narrator demo

ABasic programs: Games

BrickOut	classic computer brick wall game
Othello	also known as 'go'
Saucer	simple shoot-em-up game
Spelling	simple talking spelling game
ToyBox	selectable graphics demo

ABasic programs: Sounds

Entertainer	plays that tune
HAL9000	pretends it's a real computer
Police	simple police siren sound
SugarPlum	plays "The Dance of the Sugarplum Fairies"

C programs:

ATerm	simple terminal program, S-E
cc	aid to compiling with Lattice C
decvnt	opposite of CONVERT for cross developers
Dotty	source code to the 'dotty window demo
echox	unix-style filename expansion, partial S,O-D
fasterfp	explains use of fast-floating point math
FixDate	fixes future dates on all files on a disk, S-E
freedraw	simple Workbench drawing prog.,S-E
GfxMem	graphic memory usage indicator, S-E
Grep	searches for a given string in a file with docs.
ham	shows off the hold-and-modify method of color generation
IBM2Amiga	fast parallel cable transfers between an IBM and an Amiga
Mandel	Mandelbrot set program, S-E
moire	patterned graphic demo, S-E
objfix	makes Lattice C object file symbols visible to Wack, S-E
quick	quick sort strings routine
raw	example sample window I/O
setlace	turns on interlace mode, S-E
sparks	qix-type graphic demo, S-E

Other executable programs:

SpeechToy	speech demonstration
WhichFont	displays all available fonts

Texts:

68020	describes 68020 speedup board from CSA
Aliases	explains uses of the ASSIGN command
Bugs	known bug list in Lattice C 3.02
CLICard	reference card for AmigaDOS CLI
CLICommands	guide to using the CLI
Commands	shorter guide to AmigaDOS CLI commands
EdCommands	guide to the ED editor
Filenames	AmigaDOS filename wildcard conventions
HalfBright	explains rare graphics chips that can do more colors
ModemPins	description of the serial port pinout
RAMdisks	tips on setting up your RAM: disk
ROMWack	tips on using ROMWack
Sounds	explanation of Instrument demo sound file format
Speed	refutation of Amiga's CPU and custom chip speed
WackCmds	tips on using Wack

AMICUS Disk 2

C programs:

alib	AmigaDOS object library manager , S-E
ar	text file archive program, S-E
fixobj	auto-chops executable files
shell	simple CLI shell, S-E
sq, usq	file compression programs, S-E
YachtC	a familiar game, S-E
Make	a simple 'make' programming utility, S-E
Emacs	an early version of the Amiga text editor, S-E-D

Assembler programs:

bsearch.asm	binary search code
qsort.asm	Unix compatible qsort() function, source and C test program
setjmp.asm	setjmp() code for Lattice 3.02
SVprintf	Unix system V compatible printf()
trees.o	Unix compatible tree() function, O-D

(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)

John Draper Amiga Tutorials:

Animate	describes animation algorithms
Gadgets	tutorial on gadgets
Menus	learn about Intuition menus

AMICUS Disk 3

C programs:

Xref	a C cross-reference gen., S-E
6bitcolor	extra-half-bright chip gfx demo, S-E
Chop	truncate (chop) files down to size, S-E
Cleanup	removes strange characters from text files
CR2LF	converts carriage returns to line feeds in Amiga files, S-E
Error	adds compile errors to a C file, S
Hello	window ex. from the RKM, S
Kermit	generic Kermit implementation, flakey, no terminal mode, S-E
Scales	sound demo plays scales, S-E
SkewB	Rubik cube demo in hi-res colors, S-E

AmigaBasicProgs(dir)

Automata	cellular automata simulation
CrazyEights	card game
Graph	function graphing programs
WitchingHour	a game

ABasiC programs:

Casino	games of poker, blackjack, dice, and craps
Gomoku	also known as 'othello'
Sabotage	sort of an adventure game

Executable programs:

Disassem	a 68000 disassembler, E-D
DpSlide	shows a given set of IFF pictures, E-D
Arrange	a text formatting program, E-D

Assembler programs:

Argoterm	terminal program with speech and Xmodem, S-E

AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascii.c, idemo.c, idemo.guide, idemo.make, idemoall.h, nodos.c, and txwrite.c

addmem.c	add external memory to the system
bobtest.c	example of BOB use
consoleIO.c	console IO example
creaport.c	create and delete ports
creastdi.c	create standard I/O requests
creatask.c	creating task examples
diskio.c	example of track read and write
dotty.c	source to the 'dotty window' demo
dualplay.c	dual playfield example
flood.c	flood fill example
freemap.c	old version of 'freemap'
geltools.c	tools for VSprites and BOBs
gfxmem.c	graphic memory usage indicator
hello.c	window example from RKM
inputdev.c	adding an input handler to the input stream
joystik.c	reading the joystick
keybd.c	direct keyboard reading
layertes.c	layers examples
mousport.c	test mouse port
ownlib.c, ownlib.asm	example of making your own library with Lattice
paratest.c	tests parallel port commands
seritest.c	tests serial port commands
serisamp.c	example of serial port use
prinintr.c	sample printer interface code
prtbase.h	printer device definitions
regintes.c	region test program
setlace.c	source to interlace on/off program
setparallel.c	set the attributes of the parallel port
SetSerial.c	set the attributes (parity, data bits) of the serial port
singplay.c	single playfield example
speechtoy.c	source to narrator and phonetics demo
timedely.c	simple timer demo
timer.c	exec support timer functions
timrstuf.c	more exec support timer functions
WhichFont.c	loads and displays all available system fonts

process.i and prtbase.i assmebler include files:

autorqstr.txt	warnings of deadlocks with autorequesters
consoleIO.txt	copy of the RKM console I/O chapter
diskfont.txt	warning of disk font loading bug
fullfunc.txt	list of #defines, macros, functions
inputdev.txt	preliminary copy of the input device chapter

License information on Workbench distribution license

printer pre-release copy of the chapter on printer drivers, from RKM 1.1

v11fd.txt 'diff' of .fd file changes from version 1.0 to 1.1 v28v1.diff 'diff' of include file changes from version 28 to 1.0

AMICUS Disk 5 Files from the Amiga Link / Amiga Information Network

Note that some of these files are old, and refer to older versions of the operating system. These files are from Amiga Link. For a time, Commodore supported Amiga Link, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

A demo of Intuition menus called 'menudemo', in C source

whereis.c	find a file searching all subdirectories
bobtest.c	BOB programming example
sweep.c	sound synthesis example

Assembler files:

mydev.asm	sample device driver
mylib.asm	sample library example
mylib.i	
mydev.i	
asmsupp.i	
macros.i	assembler include files

Texts:

amigatricks	tips on CLI commands
extdisk	external disk specification
gameport	game port spec
parallel	parallel port spec
serial	serial port spec
v1.1update	list of new features in version 1.1
v1.1h.txt	'diff' of include file changes from version 1.0 to 1.1

Files for building your own printer drivers, including dospecial.c, epsondata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.

AMICUS Disk 6 IFF Pictures

This disk includes the DPSlide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon. The pictures include a screen from ArticFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Martian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Contruction Set, a TV newcaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.

AMICUS Disk 7 DigiView HAM demo picture disk

This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens. The 'seeilbm' program, to turn any screen into an IFF picture.

AMICUS Disk 8

C programs:

Browse	view text files on a disk, using menus S-E-D
Crunch	removes comments and white space from C files, S-E
IconExec	EXECUTE a series of commands from Workbench S-E
PDScreen Dump	dumps Rastport of highest screen to printer
SetAlternate	sets a second image for an icon, when clicked once S-E
SetWindow	makes windows for a CLI program to run under Workbench S-E
SmallClock	a small digital clockin a window menu bar
Scrimper	the screen printer in the fourth AC S-E

Amiga Basic Programs:

(**Note:** Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, & included here.)

AddressBook	a simple address book database
Ball	draws a ball
Cload	program to convert Compuserve hex files to binary, S-D
Clue	the game, Intuition driven
ColorArt	art drawing program
DeluxeDraw	the drawing program in the 3rd AC, S-D
Eliza	conversational computer psychologist
Othello	the game, as known as 'go'
RatMaze	3D ratmaze game
ROR	boggling graphics demo
Shuttle	draws 3D pictures of the space shuttle
Spelling	simple spelling program
YoYo	wierd zero-gravity yo-yo demo, tracks yo-yo to the mouse

Executable programs:

3Dcube	Modula-2 demo of a rotating cube
AltIcon	sets a second icon image, displayed when the icon is clicked
AmigaSpell	a slow but simple spell checker, E-D
arc	the ARC file compression prog must for telecom, E-D
Bertrand	graphics demo
disksalvage	prog. to rescue trashed disks, E-D
KwikCopy	a quick but nasty disk copy program: ignores errors, E-D
LibDir	lists hunks in an object file E-D
SaveILBM	saves any screen as IFF pic.E-D ?? ScreenDump shareware screen dump prog, E only
StarTerm	version 2.0, term program, XmodemE-D

Texts:

LatticeMain	tips on fixing _main.c in Lattice
GDiskDrive	make your own 5 1/4 drive
GuruMed	explains the Guru numbers
Lat3.03bugs	bug list of Lattice C version 3.03
MForgeRev	user's view of the MicroForge HD
PrintSpooler	EXECUTE-based print spool prog.

.BMAP files:

These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'clist', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathffp', mathieeedoubas', 'mathieeesingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.

AMICUS Disk 9

Amiga Basic Programs:

FlightSim	simple flight simulator program
HuePalette	explains Hue, Saturation, & Intensity
Requester	ex. of requesters from Amiga Basic
ScrollDemo	demonstrates scrolling capabilities
Synthesizer	sound program
WorldMap	draws a map of the world

Executable programs:

Boing!	latest Boing! demo,with selectable speed,E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time & date
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

Texts:

FnctnKeys	read function keys from Amiga Basic
HackerSln	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
Boing!	latest Boing! demo,with selectablespeed, E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	Intuition-based way to set the time date,
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

Texts:

FnctnKeys	explains how to read function keys from Amiga Basic
HackerSln	explains how to win the game 'hacker'
Ist68010	guide to installing a 68010 in your Amiga
PrinterTip	sending escape sequences to your printer
StartupTip	tips on setting up your startup-sequence file
XfrmrReview	list of Transformer programs that work

Printer Drivers:

Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.

AMICUS Disk 10 Instrument sound demos

This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a boink, a calliope, a car horn, claves, water drip, electric guitar, a flute, a harp arpegio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibrophone, a violin, a wailing guitar, a horse whinny, and a whistle.

AMICUS Disk 11

C programs

dirutil	Intuition-based, CLI replacement manager S-E
cpri	shows and adjusts priority of CLI processes, S-E
ps	shows info on CLI processes, S-E
vidtex	displays Compuserve RLE pics, S-E

AmigaBasic programs

pointered	pointer and sprite editor program
optimize	optimization ex ample from AC article
calendar	large, animated calendar, diary and date book program
amortize	loan amortizations
brushtoBOB	converts small IFF brushes to AmigaBasic BOB OBJECTS
grids	draw and play waveforms
hilbert	draws Hilbert curves
madlib	mad lib story generator
mailtalk	talking mailing list program
meadows3D	3D graphics program, from A C™ article
mousetrack	mouse tracking example in hires mode
slot	slot machine game
tictactoe	the game
switch	pachinko-like game
weird	makes strange sounds

Executable programs

cp	unix-like copy command, E
cls	screen clear, S-E
diff	unix-like stream editor uses 'diff' output to fix files
pm	chart recorder performances indicator

Assembler programs

cls	screen clear and CLI arguments example

Modula-2

trails	moving-worm graphics demo
caseconvert	converts Modula-2 keywords to uppercase
Forth	Breshehan circle algorithm example
Analyze	12 templates for the spreadsheet Analyze

There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.

AMICUS Disk 12

Executable programs

blink	'alink' compatible linker, but faster, E-D
clean	spins the disk for disk cleaners, E-D
epsonset	sends Epson setttings to PAR from menu E-D
showbig	view hi-res pics in low-res superbitmap, E-D
speaktime	tell the time, E-D
undelete	undeletes a file, E-D
cnvapldhm	converts Apple][low, medium and high res pictures to IFF, E-D
menued	menu editor produces C code for menus, E-D
quick	quick disk-to-disk nibble copier, E-D
quickEA	copies Electronic Arts disks, removes protection, E-D
txed 1.3	demo of text editor from Microsmiths,E-D

C programs

spin3	rotating blocks graphics demo, S-E-D
popcli	start a new CLI at the press of a button, like Sidekick, S-E-D
vsprite	VSprite example code from Commodore, S-E-D
AmigaBBS	Amiga Basic bulletin board prog., S-D

Assembler programs

star10	makes star fields like Star Trek intro,S-E-D

Pictures

Mount Mandelbrot	3D view of Mandelbrot set
Star Destroyer	hi-res Star Wars starship
Robot	robot arm grabbing a cylinder

Texts

vendors	Amiga vendors, names, addresses
cardco	fixes to early Cardco memory boards
cinclude	cross-reference to C include files
mindwalker	clues to playing the game well
slideshow	make your own slideshows from the Kaleidoscope disk

For PDS orders, please use form on page CIII
Visa and MasterCard is available on orders of $20.00 or more.

AMICUS Disk 13

Amiga Basic programs

Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs

gravity	Sci Amer Jan 86 gravitation graphic simulation, S-E-D

Texts

MIDI	make your own MIDI instrument interface, documentation & a hi-res schematic.

AMICUS Disk 14

Several programs from Amazing Computing issues:

Tools

Dan Kary's	C structure index program, S-E-D

Amiga Basic programs:

BMAP Reader	by Tim Jones
IFFBrush2BOB	by Mike Swinger
AutoRequester	example
DOSHelper	Windowed help system for CLI commands, S-E-D
PETrans	translates PET ASCII files to ASCII files, S-E-D
C Squared	Graphics program from Scientific American, Sept 86, S-E-D
crlf	adds or removes carriage returns from files, S-E-D
dpdecode	decrypts Deluxe Paint, remo
ves copy	protection, E-D
queryWB	asks Yes or No from the user returns exitt code, S-E
vc	VisiCalc type spreadsheet, no mouse control, E-D
view	views text files with window and
slider	gadget, E-D

Oing, Sproing, yaBoing, Zoing are sprite-based
Boing! style demos, S-E-D

CLIClock, sClock, wClock are window border clocks, S-E-D

Texts

An article on long-persistance phospor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.

AMICUS Disk 15

The C programs include:

'pr'	a file printing utility, which can print files in the background, and with line numbers and control character filtering.
'fm'	displays a chart of the blocks allocated on a disk.
'Ask'	questions an 'execute' file, returns an error code to control the execution in that batch file
'Stat'	an enhanced version of AmigaDOS 'status' command.
'Dissolve'	random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.
'PopCLI2'	invoke new CLI window at the press of a key.

The executable programs include:

'Form'	file formatting program through the printer driver to select print styles
'DiskCat'	catalogs disks, maintains, sorts,merges lists of disk files
'PSound'	SunRize Industries' sampled sound editor & recorder
'Iconmaker'	makes icons for most programs
'Fractals'	draws great fractal seascapes and mountain scapes.
'3D Breakout'	3D glasses, create breakout in a new dimension
'AmigaMonitor'	displays lists of open files,
memory use,	tasks, devices and ports in use.
'Cosmoroids'	version of 'asteroids' for the Amiga.
'Sizzlers'	high resolution graphics demo writteN in Modula 2.

Texts:

'ansi.txt'	explains escape sequences the CON: device responds to.
'FKey'	includes template for making paper to sit in the tray at the top of the Amiga keyboard.
'Spawn'	programmer's document from Commodore

Amiga, describs ways to use the Amiga's multitasking capabilities in your own programs.

AmigaBasic programs:

'Grids'	draw sound waveforms, and hear them played.
'Light'	a version of the Tron light-cycle video game.
'MigaSol'	a game of solitaire.
'Stats'	program to calculate batting averages
'Money'	"try to grab all the bags of money that you can."

AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.

AMICUS Disk 16

'juggler' demo by Eric Graham. a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.

IFF pictures

parodies of the covers of Amiga World and Amazing Computing.

C programs:

'Inputhandler'	example of making an input handler.
'FileZap3'	binary file editing program
'ShowPrint'	displays IFF picture, and prints it.
'Gen'	program indexes and retrieves C structures and variables declared in the Amiga include file system.

Executable Programs:

'FixHunk2'	repairs an executable program file for expanded memory
'ms2smus'	converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.
'Missile'	Amiga version of the 'Missile Command' video game,

This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park

AMICUS Disk 17

Telcommunications disk whuch contains six terminal programs.

"Comm" V1.33	term prog. with Xmodem, WXmodem,
"ATerm" V7.2	term prog. includes Super Kermit
"VT-100"V2.6	Dave Wecker's VT-100 emulator with Xmodem,Kermit, and scripting
"Amiga Kermit"	V4D(060) port of the Unix C-Kermit
"VTek" V2.3.1	Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression
"AmigaHost"	V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol.
"FixHunk"	expansion memory necessity
"FixObj"	removes garbage characters from modem received files
"Txt"	filters text files from other systems to be read by the Amiga E.C.
"addmem"	executeable version for use with mem expansion article in AC v2..1
'arc'	file documentation and a basic tutorial on un 'arc'ing files
"arcre"	for makeing "arc" files E.C.

AMICUS Disk 18

Logo	Amiga version of the popular computer language, with example programs, E-D
Tv*Text	Demo version of the TV*Text character generator
PageSetter	Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package.
FullWindow	Resizes any CLI window using only CLI commands, E-D
Life3d	3-D version of Conway's LIFE program, E-D
Defdisk	CLI utility to re-assign a new Workbench disk, S-E-D
Calendar.WKS	Lotus-compatible worksheet that makes calendars
SetKey	Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D
VPG	Video pattern generator for aligning monitors, E-D
HP-10C	Hewlett-Packard-like calculator, E-D
SetPrefs	Change the Preferences settings on the fly, in C, S-E-D
StarProbe	Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D
ROT	C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D
Scat	Like Ing, windows on screen run away from the mouse, E-D
DK	Decays" the CLI window into dust, in Modula 2, S-E-D
DropShadow2	Adds layered shadows to Workbench windows, E-D

AMICUS Disk 19

This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffith, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.

Solve	Linear equation solver in assembly language, S-E-D
Gadgets	Bryan Catley's AmigaBasic tutorial, S-D
Household	Bryan Catley's AmigaBasic household inventory program, S-D
Waveform	Jim Shields' Waveform Workshop: AmigaBasic, S-D
DiskLib	John Kennan's AmigaBasic disk librarian program, S-D
Subscripts	Ivan Smith's AmigaBasic subscript example, S-D
String, Boolean	C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D
Skinny C	Bob Riemersma's example for making small C programs, S-E-D
COMAL.h	Make C look like COMAL header file, S-D
EmacsKey	Makes Emacs function key definitions by Greg Douglas, S-D
AMon 1.1	Snoop on system resource use, E-D
BTE	Bard's Tale character editor, E-D
Size	CLI program shows the size of a given set of files, E-D
WinSize	CLI window utility resizes current window, S-E-D

AMICUS Disk 20

Compactor, Decoder	Steve Michel AmigaBasic tools, S-D
BobEd	BOB and sprite editor written in C,S-E-D
SpriteMasterII	Sprite editor and animator by Brad Kiefer, E-D
BlitLab	Blitter chip exploration C program by Tomas Rokicki, S-E-D
FPic	Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D
Bankn	Complete home banking prog., balance your checkbook! E-D

AMICUS Disk 21

Target	Makes each mouse click sound like a gunshot, S-E-D
Sand	Simple game of sand that follows the mouse pointer, E-D
PropGadget	Harriet Maybeck Tolly's proportional gadget example, S-E
EHB	Checks to see if you have extra-half-bright graphics, S-E-D
Piano	Simple piano sound program
CelScripts	Makes cel animation scripts for Aegis Animator, in AmigaBasic

This disk has electronic catalogs for AMICUS disks 1 to 20 and Fish disks 1 to 80. They are viewed with the DiskCat program, included here.

AMICUS Disk 22

Cycles	Light cycle game, E-D
Show_PrintII	Views and prints IFF pictures, including larger than screen
PrtDrvGen2.3	Latest version of a printer driver generator
Animations	VideoScape animations of planes and boing ball
Garden	Makes fractal gardenscapes
BasicSorts	Examples of binary search and insertion sort in AmigaBasic

AMICUS Disk 23

An AMICUS disk completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing "Big Sound" on your Amiga

Instruments	a collection of 25 instruments for playing and creating music. The collection ranges from Cannon to Marimba
List INSTR	program to list the instruments DMCS will not load as well as list the origins for any instrument.
Music	a collection of 14 Classical pieces
1812Overture	The 16 minute classical feature complete with Cannon!

Three Amiga Music Players:
SMUSPlay
MusicCraft2SMUS
MusicStudio2SMUS

AMICUS Disk 24

Sectorama	A disk sector editor for any AmigaDOS file-structured device, recover files from a trashed hard disk. By David Joiner of MicroIllusions
Iconize	Reduces the size of IFF images, companion program, Recolor, remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, make icons look like miniatures of the pictures.
CodeDemo	Modula-2 program converts assembler object files to inline CODE statements. Comes with a screen scrolling example
AmiBug	Workbench hack makes the same fly walk across the screen at random intervals. Otherwise, completely harmless.
BNTools	Three examples of assembly language code from Bryce Nesbitt: 1. SetLace,prog to switch interlace on&off. 2. Why, replace AmigaDOS CLI Why 3. LoadIt, prog to load a file into memory until a reboot. (Only the most esoteric hackers will find LoadIt useful.)
Monolace	CLI program resets Preferences to several colors of monochrome & interlace screens. C source is included, works with DisplayPref, a CLI program which displays the current Preferences settings.
BoingMachine	A ray-traced animation of a perpetual motion Boing-making machine, includes the latest version of the Movie program, which has the ability to play sounds along with the animation. By Ken Offer
Daisy	Example of using the translator and narrator devices to make the Amiga talk. It is written in C.
QuickFlix	Script-driven animation and slideshow program flips through IFF images.
BMon	System monitor AmigaBasic program ; perform simple manipulations of memory.
Moose	Random background program, a small window opens with a moose resembleing Bullwinklesaying witty phrases user definable.
DGCS	Deluxe Grocery Construction Set, simple Intuition-based prog for assembling and printing a grocery list.
The Virus Check	directory holds several programs relating to the software virus that came to the US from pirates in Europe as detailed in Amazing Computing V2.12. Bill Koester's full explanation of the virus code is included. One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks.

AMICUS Disk 25

Nemesis	Graphics demo pans through space towards the mythical dark twin of the sun with wonderful music and space graphics.
The KickPlay	directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADDMEM for old expansion memory, as well as the ability to change the picture of the "Insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk.
KeyBird	BASIC prog edits keymaps, adjust the Workbench keymaps or create your own.
8ColorWB	Modifies the Workbench so three bitplanes are used, icons can have eight colors, instead of four, eight-color icons are included. Public domain program "zapicon" or "brush2icon" converts eight-color IFF brushes to icons, to use Deluxe Paint to make icons for this new Workbench.
BrushIcon	Converts brushes to icons (bizarr docs).
Egraph	Graphing prog reads [x,y] values from a file and displays them on the screen, similar to the same-named Unix program.
Keep 1.1	Message-managing program for telecommunications, lets you save messages from an online transcript to another file, understands the message format of the national networks and several types of bulletin board software. Moves through the transcript and save messages
Kill.fastdir	Speed up directory access, it creates a small file in each directory on a disk which contains the information about the files, will also remove all the "fastdir" files from each directory. by CLImate's authors
The LaceWB	program changes between interlace and non-interlace Workbench. Previously, you were forced to reboot after changing Preferences to an interlaced screen. This program flips between the normal and extended screen heights.
PW_Utility	A shareware utility for ProWrite users, changes margin settings and font types.
Guru	A CLI program, prints out probable causes for Guru mediations; C source included.
DiskWipe	Latest from Software Distillery, removes files from directories or disk drives, much faster than "delete."
Snow	AmigaBasic makes snowflake designs.
Mlist	Mailing list database.
Softballstats	Maintain softball statistics/ team records.
Dodge	Short Modula-2 program moves the Workbench screen around after a period of time, prevents monitor burn-in.

AMICUS Disk 26

Todor Fay's	SoundScape module code from his Amazing Computing articles. The source to Echo, Chord, TX, and VU is included. The Lattice and Manx C source code is here, along with the executable modules.
Claz2	Update of prog to convert IFF images to PostScript files for printing on laser printers
SDBackup	Hard disk backup prog with Lempel-Ziv compression to reduce the necessary number of disks.
TCB	Prints information about tasks and processes in the system; assembler source is included.
FunBut	Lets a function key act like a rapid series of left mouse button events.
DC	A handy program for people who use an Amiga 1020 5 1/4 inch drive as an AmigaDOS floppy. A Workbench program that sends a DiskChange signal to the operating system: instead of typing "diskchange df2:" over and over again, just click on the icon. C source included.
System config	File makes screen 80 columns wide of text in the Scribble! word processor.
Dick2Ram	2 programs to move the Scribble! spelling dictionary to and from the RAM disk.
Lexical	Analyzes a text file and gives the Gunning-Fog, Flesch, and Kincaid indices which measure readability.
HexDump	Modula-2 program to display memory locations in hexadecimal.
Tartan	AmigaBasic; design Tartan plaids.
DirMaster	Disk catalog program.
BMP	plays 8SVX sampled sounds in the background while something else is happening in the Amiga, as your Amiga is booting, for example.
ShowPt	CLI program changes your pointer to a given pointer.

AMICUS 26 also has a collection of mouse pointers, & Workbench program to display them

SORRY...

Do to the ever increasing size of the Fred Fish Collection, AC will no longer be able to maintain a complete catalog of Public Domain Software in every issue. However, AC will continue to update this abrievated catalog with new additions to the library. A full catalog of every disk and program in the Amicus and Fred Fish Libraries is available in the current issue of:

AC's GUIDE AMIGA

At your local Amazing Dealer

Fred Fish Public Domain Software

Fred Fish Disk 179

DietAid	Diet planning aid to allow the user to compile lists of ingredients (recipes) and automatically compute calorie totals, etc. Update FF36. V3.1, binary only. by Terry Gintz
Dmake	Beta release of Matt's version of the UNIX make utility. Features multiple dependancies, wildcard support, and more. Includes source. By Matt Dillion
Excption	Exception is a set of error handling routines that provide a programmer with the ability to easily handle often difficult to implement routines. Routines such as no more memory, file not open, read/write error....etc. V0.6, includes source. By Gerald T Hewes
KickFont	For A-1000 owners, will permanently replace the topaz font on the kickstart disk with a font called "look". Includes a sample in the form of an IFF picture. V3.0, binary only. Also included is Benjamin Fuller's freely redistributable 'SumKick' program. By Greg Browne
Launch	Sample program showing how you can load and execute a program in the workbench environment, then return to the CLI. Includes source. By Peter da Silva
Regexp	A nearly-public-domain reimplementation of the V8 regexp(3) package. Gives C programs the ability to use egrep-style regular expressions, and does it in a much cleaner fashion than the analogous routines in SysV. Includes source. By Henry Spencer
TSnip	Very nice "cut and paste" type utility with lots of uses and functions. Features a pop-up intuition control panel, multiple font and color recognition, clipboard and pipe support and a couple of utility programs. V1.4a, source for support programs only. By John Russell
UnixUtil	A few CLI utilities, including some functionally similiar to the UNIX utilities of the same names. Included are: Wc, Head, Tail, Tee, Detab, Entab, and Trunc. Descriptions are given in the included '.doc' files. By Gary Brant

Fred Fish Disk180

Browser	A programmer's "Workbench". Allows you to easily and conveniently move, copy, rename, and delete files & directories from a CLI environment. Also provides a method to execute either Workbench or CLI programs. V1.6, update to FF134, binary only. By Peter da Silva
GeoTime	A couple of interesting "clock" type programs based on the "Geochron". Observe the earth's shadow scroll across a map or globe in real-time, based on the system clock. V1.0, binary only, shareware. By Mike Smithwick
GPrint	A black & white graphics print utility for Epson compatible printers. Command-line options allow several different print qualities and densities. Includes a couple of sample IFF files for printing. V2.03, binary only, shareware. By Peter Cherna
Jed	A nicely done, intuition-based editor that is quite user-friendly. Features word-wrap, auto-indent, newcli, alt buffer, split-window, keyboard macro, help, printing, and more. V1.0, binary only, shareware. By Dan Burris
NoVirus	Another Anti-Virus utility. This one features known and new virus detection, view boot block, save and restore bootblocks, several "Install" options and more. Written in assembly. V1.56, binary only. By Nic Wilson
RepString	Nice little CLI utility to replace any type of string in any type of file with another string of any type. V1.0, binary only, shareware. By Luciano Bertato
TrekTrivia	Very nice mouse-driven trivia type program for Star Trek fans. Contains 100 questions with additional trivia disks available from the author. Binary only, shareware. By George Broussard

Fred Fish Disk 181

AMXLISP	Amiga-ized version of the XLisp interpreter originally by David Betz. V2.00, includes source. By David Betz; Amiga work by Francois Rouaix
Bally	Amiga port of the former arcade game named Click. Lacks sound effects,promised for later updates. V0.1, binary only, shareware. By Oliver Wagner
Tracker	Useful debugging routines similar in function but more versatile to those of "MemTrace" on FF163. Will track and report on calls to AllocMem(), FreeMem() [or lack thereof!] among others. V0.0a (Alpha release). By Karl Lehenbauer

Fred Fish Disk 182

AMC	"Amiga Message Center". Scrolls a message from a text file across the screen on a colorful background. Similar to the "greetings" programs developed by European Amiga enthusiasts. V1.0, binary only. By Foster Hall
Edimap	A keymap editor. Allows you to read in an existing keymap file, modify it to suit your needs, and save it as a ready-to-use keymap. V1.0, includes source. Author: Gilles Gamesh
HR136	An IFF file containing a chart showing every possible mixture of the sixteen basic palette colors. Also included are optimized and monochrome palettes along with several tips and techniques for using them with various paint programs.By Dick Bourne
Iconmerger	Intuition-based program to take any two brush files and merge them into an alternate-image type icon. V2.0, binary only. By Terry Gintz
Sam	Another IFF sound player with several command-line options. Includes several samples. V1.0, binary only. By Nic Wilson
SetFont	Allows you to change the system font with various command-line options. Cleans up all known bugs in FF75. V2.5, includes source in C++. By Dave Haynie

Fred Fish Disk 183

FixFd	A utility for Amiga assembly programmers. FixFd will read a '.FD' file and output a file that can be 'INCLUDE'ed rather than having to link with the colossal 'Amiga.Lib'. V1.0, includes source in assembly. By Peter Wyspianski
Mklib	Another example of building a shared library that evolved from "Elib" FF87. Also included is a library, Edlib, which contains several functions not included in the Manx standard libraries. Includes source. By Edwin Hoogerbeets with C-functions from several different authors

For PDS orders, please use form on page CIII
Visa and MasterCard is available on orders of $20.00 or more.

PCQ A subset implementation of a freely-redistributable Pascal compiler. Supports include files, external references, records, enumerated types, pointers, arrays, strings and more. Presently does not support range types, the 'with' statement or sets. V1.0, includes source and sample programs. By Patrick Quaid

Fred Fish Disk 184

BI A small brush to to C-code image converter, intended to be used from CLI. V1.0, binary only. By Terry Gintz

CardMaker A programmer's aid for creating card image data that can be used in any card game that uses the standard 52 card deck. V1.0, binary only. By Terry Gintz

DPS Demo version of a program that will allow you to take any IFF file and save it as a totally self-contained executable file, without the need for any IFF-viewers. V1.0, binary only. By Foster Hall

MouseUtil Intuition based program to allow you to change your mouse speed without having to go through preferences. V1.1, includes assembly source. By Luciano Bertato

Print Small print utility designed to replace the "copy <filename> to prt:" command. Opens a window displaying the filename being printed, length, and a status bar showing percent completed. Also includes an abort gadget. V1.0, binary only. By Luciano Bertato

VacBench This amusing little screen hack will "clean up" your WorkBench screen for you when it gets too cluttered! Binary only. By Randy Jouett

World A text adventure game similar to the Infocom adventures of Planetfall and Starcross. Quite large with a tremendous variety of responses. V1.02, includes source. By Doug Mcdonald, Amiga port by Eric Kennedy

Fred Fish Disk 185

Commodore IFF This is a copy of the official November 1988 Commodore IFF disk. All the files in the "documents" directory are in zoo file "documents.zoo"

Fred Fish Disk 186

A68k A 68000 assembler originally written in Modula-2 in 1985 and converted to C by Charlie Gibb in 1987. Has been converted to accept metacomco-compatible assembler source code and to generate Amiga objects. Includes source. This is V2.42, an update to FF110. By Brian Anderson; C translation and Amiga work by Charlie Gibb

Cards'O'Rama A simple game that let's you push your memory. It is played with a deck of 32 cards, grouped in 16 pairs. The cards are shuffled and then displayed at the beginning of each game. Your goal is to pick up as many pairs as you can, until there are no cards left on the screen. V1.0, includes source. By Werther Pirani

Qt2 A cute program that gives the time the way many people actually do, I.E. "it's nearly ten to five". Includes source in assembly. By Charlie Gibb

SimCPM A CP/M simulator for the Amiga. Simulates an 8080 along with H19 terminal emulation. Includes source. This is V2.3, an update to FF109. By Jim Cathey; Amiga port by Charlie Gibbs and Willi Kusche

Fred Fish Disk 187

Diskperf A disk benchmark program which runs on both Unix and the Amiga. This is an update to FF48, with bug fixes and more reliable measurements of the faster read and write speeds available under the new Fast File System. By Rick Spanbauer, enhancements by Joanne Dow

HackLite This is the latest version the Amiga port of Hack, with lots of Amiga specific enhancements and neat graphics. Now includes an easy to use installation program. This is HackLite V1.0.0, binary only. By Software Distillery

Mackie A versatile cli/macro-key initiator based on POPCLI with a unique method of "screen-blanking". I won't say more, just try it! V 1.13, includes source. This is an update to FF161. By Tomas Rokicki

SetCPU A program designed to allow the user to detect and modify various parameters related to 32 bit CPUs. Includes commands to enable or disable the text/data caches, switch on or off the '030 burst cache line fill request, use the MMU to run a ROM image from 32-bit memory, and to report various parameters when called from a script. V1.4, includes source. By Dave Haynie

Fred Fish Disk 188

BootIntro This program creates a small intro on the bootblock of any disk, which will appear after you insert the disk for booting. The headline can be up to 20 characters. The scrolling text portion can be up to 225 characters. V1.0, binary only. By Roger Fischlin

DiffDir Diffdir compares the contents of two directories, reporting on differences such as files present in only one directory, different modification dates, file flags, sizes, comments, etc. V1.0, includes source. By Mark Rinfret

ExecDis A disassembler comment generator program for the 1.2 Kickstart ROM exec library image. Generates a commented disassembly of the exec library. V1.0, binary only. By Markus Wandel

FastGro A fractal program, simulating Diffusion-Limited Aggregation (DLA) as described in the December 1988 Scientific American in the Computer Recreations column. This program is about an order of magnitude faster than the "SLO GRO" program described in Scientific American. V 1.0, includes source. By Doug Houck

FracGen A fractal generator program that generates fractal pictures from "seeds" that you create. This is unlike any of the other "fractal generators" I've seen. It can be used to load and display previously created fractal pictures, modify existing fractals, or create your own fractals. V 1.23, binary only, update to FF142. By Doug Houck

MemoryClock A clock program that shows the amount of free fast ram, free chip ram, as well as the time and date. Includes source in assembly code. By Roger Fischlin

MinRexx A simple ARexx interface which can be easily patched into almost any program. Includes as an example the freedraw program from FF1. Includes source. By Tomas Rokicki

Null A new dos device that behaves like "NIL:" but unlike "NIL:", it is a real handler. This makes it useful in lots of situations where "NIL:" cannot be used. V 0.0, includes source. By Gunnar Nordmark

TextDisplay A text display program, like "more" or "less", but about half the size and handles all screen formats (pal/ntsc, interlace/non-interlace, etc). V 1.1, binary only. By Roger Fischlin

Fred Fish Disk 189

Mackie A versatile cli/macro-key initiator based on POPCLI wit a unique method of "screen-blanking". I won't say more, just try it! Version 1.20, includes source. update of FF187. Author: Tomas Rokicki

NetHack This is part 1 of a two part distribution of NetHack, which was too large to fit on a single disk, even when zoo'd. Part 2 is on disk 190. Both parts, along with zoo to unpack them, are required to use or rebuild NetHack. V 2.3. Includes source. Author: Various; Amiga work by Olaf Seibert

Uedit V 2.4g shareware editor. Has learn mode, a command language, menu customization, and other user configurability and customizability features. Binary only, shareware, Update to FF 173 Author: Rick Stiles

Fred Fish Disk 190

GaryIcons A collection of more interesting and useful icons. Author: Gary Roseman

LBM2Image Takes an IFF picture and generates a C source module which can be compiled and linked with your program to display the picture with the intuition DrawImage function. Binary only. by: Denis Green

NetHack This is part 2 of a two part distribution of NetHack, which was too large to fit on a single disk, even when zoo'd. Part 1 is on disk 189. Both parts, along with zoo to unpack them, are required to use or rebuild NetHack. V 2.3 Includes source. Author: Various; Amiga work by Olaf Seibert

Fred Fish Disk 191

BlitLab Blitlab is a program which lets you experiment with the blitter, to your hearts content, in relative safety. It opens a workbench window with gadgets for all the registers of the blitter, and allows you to manipulate individual registers and perform blits on a magnified bitmap. V 1.4, an update to FF84. Includes source. Author: Tomas Rokicki

Blk A requester making tool employing various recursive algorithms including a recursive parser. It takes input text files and converts them to C-source for including as requester declarations. Update to FF152, with many enhancements. Includes source. Author: Stuart Ferguson

FileBootBlock This simple little program reads blocks 0 and 1 of a bootable disk and saves them as a program file that can be run (heaven forbid) or disassembled by programs like DIS or DSM. Includes source in assembly code. by: John Veldthuis

Spell A port of a Unix version of a freely distributable screen oriented, interactive, spelling checker. Update to FF54, with enhancements by Tomas Rokicki. V2.0.02, includes source. Author: Pace Willisson; enhancements by Tomas Rokicki

Pz15 Computer version of those cheap plastic puzzles with 15 white tiles numbered 1 through 15 and an empty square in a 4 by 4 arrangement. This one is more challenging since you can't solve it by just prying out the pieces. Includes source. Author: Mike Hall

Fred Fish Disk 192

Eval This package allows you to manipulate expressions. Currently its two main functions are evaluation and differentiation. It also does some basic simplifications (based on pattern matching) to make the result of a differentiation more presentable. Includes source. Author: David Gay

PacMan87 This is a nice little "pacman like" game with some new features like fire pits, stabbing knives, electric arcs and flame throwers, that must be avoided. Has three levels of difficulty, easy, medium, and hard. Sounds can be toggled on or off. Keeps a record of the top ten scores. Shareware, binary only. Author: Steve Jacobs and Jim Boyd

ReSourceDemo A demo version of ReSource, an interactive disassembler for the Amiga. This is a complete version except that the "save" features have been disabled. V0.36, binary only. by: Glen McDiarmid

Fred Fish Disk 193

KeyMapEd Allows you to change the KeyMaps used with SetMap This is a full featured editor providing support for normal, string and dead keys. The keyboard represented is from an A2000/A500 but it is fully compatible with A1000 keyboards. V1.02, includes source Author: Tim Friest

Zc This is a modified version of the Sozobon C compiler fromFF171. It has been modified to generate code compatible with the A68k assembler from FF186 and a new frontend control program makes it easy to use like the UNIX "cc" frontend. V1.01, includes source. by: Johann Ruegg; Amiga work by Joe Montgomery

Fred Fish Disk 194

Moria A single player dungeon simulation. The object of the game is to defeat the Balrog, which lurks in the deepest levels of the dungeon. You begin at the town level above the dungeon, where you may acquire supplies, weapons, armor, and magical devices by bartering with various shop owners, before descending into the dungeon to do battle. Amiga enhancements include pull down menus, graphics mode, pickup mode, a continuous move mode, a real time mode, a message wait time mode, as well as other modifications to improve overall playability and to take advantage of the unique features of the Amiga. V3.0, binary only, requires at least 1Mb of memory. Author: Robert Alan Koeneke and others. Amiga version by Richard Henderson and others.

Fred Fish Disk 195

MicroEMACS Version 3.10 of Daniel Lawrence's variant of Dave Conroy's microemacs. This is an update to the version released on disk 119. New features include multiple marks, more function key support, a better crypt algorithm, and end-of-word command, a command line switch for setting environment variables, new hooks for macros, a command to strip trailing whitespace, international-zation features like foreign language message support, horizontal window scrolling, much faster search algorithm, Amiga intuition support, and more. Includes source and extensive online documentation. Author: Dave Conroy, MANY enhancements by Daniel Lawrence

Fred Fish Disk 196

HamPics These are some of the most stunning digitized pictures yet for the Amiga. They were scanned at a resolution of 4096 by 2800 pixels, 36-bits per pixel, on an Eikonix 1435 slide scanner, cropped, gamma corrected, scaled, and converted to Amiga IFF HAM files. They are displayed with a special ILBM loader that handles overscan HAM images. Includes source for the display program. Author: Jonathan Hue

Fred Fish Disk 197

CTags Create a tags file from the specified C, Pascal, Fortran, YACC, lex, or lisp sources. A tags file can be used by a cooperating editor to quickly locate specified objects in a program's source code. Berkeley V4.7, includes source. Author: Ken Arnold, Jim Kleckner, and Bill Joy Ported to Amiga by G. R. (Fred) Walter

Find Find is a utility which searches for files that satisfy a given boolean expression of attributes, starting from a root pathname and searching recursively down through the hierarchy of the file system. Very much like the Unix find program. V1.2, includes source. Update to FF134 Author: Rodney Lewis

FixHunk A program to modify executable files to allow them to run in external memory. It forces all DATA and BSS hunks in the file to be loaded into CHIP memory. CODE hunks will still load into FAST ram if available. New features include an interactive mode to select where each DATA or BSS hunk will load into memory, support for overlays, support for AC BASIC compiled programs, and support for new hunk types as used by "blink". V2.1, binary only. Update to FF36. Author: D.J. James

Nro Another roff style text formatter. This is version 1.5, an update to the version released on disk 79. New features include generation of ANSI/ISO codes for bold, italics, and underline, more than one formatting command on a line, longer macro names, and many more formatting commands. Includes source. Author: Unknown, posted to usenet by Alan Vymetalik Many enhancements by Olaf Seibert

Stevie A public domain clone of the UNIX 'vi' editor. Supports window-sizing, arrow keys, and the help key. V3.35a, includes source. Update to FF166. Author: Various, Amiga work by G. R. (Fred) Walter

Fred Fish Disk 198

Charon Charon is Bradley's entry for the First Annual Badge Killer Demo Contest. The text of the demo was written by Lord Dunsany (long before the Amiga). Bradley created the illustrations and animation. The sound track is a traditional Scottish tune "The Arran Boat". by: Lord Dunsany (1915), Bradley Schenck (1988)

Fred Fish Disk 199

ASimplex An implementation of the Simplex algorithm for solving linear programs. It uses the standardized MPSX-format for input data files. V1.2, includes source. Author: Stefan Forster

Csh V3.02a of a csh like shell derived from Matt Dillon's shell, V2.07. Includes many new or improved commands, some bug fixes, etc. Includes source. Author: Matt Dillon, Steve Drew, Carlo Borreo, Cesare Dieni

MIDIsoft A program to transfer sound samples between the Amiga and a Roland S-220. V1.0, binary only. Author: Dieter Bruns

Pyro A screen blanking program that goes beyond the normal blanking process. When there are no input events, pyro takes over and starts a little fireworks display in color. V1.1, binary only. Author: Steve Jacobs and Jim Boyd

SnipDemo Demo version 1.23 of signal processing program sold by Digital Dynamics. Binary only. Author: John Hodgson

Viewer A very small program for displaying IFF pictures of any resolution. This one is written in assembly code and is only 988 bytes long. Binary only. Author: Mike McKittrick

Fred Fish Disk 200

NotBoingAgain Dr. Gandalf's entry for the First Annual Badge Killer Demo Contest. It is an interlaced HAM animation with nicely integrated sound effects. It is a great visual pun on the original Boing demo, but to say anymore would ruin the effect. Binary only, requires 1 Mb of memory. Author: Dr. Gandalf (Eric J. Fleischer, MD)

Tank This is Vincent's entry for the First Annual Badge Killer Demo Contest. It is an animation of a "fishtank simulator", with sound effects and a cute twist. Binary only. Author: Vincent H. Lee

Fred Fish Disk 201

Draco Update to Chris Gray's Draco distribution for the Amiga.Enhancements include support for floating point, register variables, more optimization, improved call/returnstandard, etc. V1.2, an update to FF76. Requires documentation from FF77 to complete the distribution kit. Binary only. Author: Chris Gray

DropCloth DropCloth lets you place a pattern, a 2 bitplane IFF image or a combination of a pattern and image, into the WorkBench backdrop. This is version 2.4, an update to version 2.2 on disk 128. Shareware, binary only. Author: Eric Lavitsky

Fred Fish Disk 202

SlavicFonts A whole bunch of new fonts from Robin LaPasha. Version 1.0. Author: Robin LaPasha

Vlt VLT is both a VT100 emulator and a Tektronix (4014 plus subset of 4105) emulator, currently in use at SLAC (Stanford Linear Accelerator Center). Although the VT100 part was originally based on Dave Wecker et al.'s VT100, many enhancements were made. The program requires ARP, and it has an ARexx port. XMODEM 1K/CRC and Kermit protocol support also included. Version 3.656, binary only. Author: Willy Langeveld

Fred Fish Disk 203

Examples Assembly and C code examples, including some old favorites (like speechtoy and yachtc3) downcoded to assembly language. Includes a replacement for the official audio device, an example of creating a subtask, a rewrite in assembly of R. J. Mical's file requester, an example of installing a custom input handler ahead of intuition, and more. Author: Jim Fiore & Jeff Glatt

GurusGuide The source files for all examples published in the "Guru's Guide, Meditation #1: Interrupts" by Carl Sassenrath, the architect of the Amiga's low-level multitasking operating system and designer of Exec. Author: Carl Sassenrath

Isam A library of routines to access relational data base systems using the Index Sequential Access Method (ISAM). This is beta version 0.9, binary only. Author: Kai Oliver Ploog

Fred Fish Disk 204

FileReq A simple file requestor, written as an exercise by the author to see how easy it would be (it wasn't). Includes source. Author: Jonathan Potter

GnuGrep The grep program from the GNU project. Replaces grep fgrep, egrep, and bmgrep. Currently does not expand Amiga style wildcards, so if you wish to scan multiple files you will need to use it with a shell that does this for you. Version 1.3, includes source. Author: Mike Haertel, James Woods, Arthur Olson, Richard Stallman, Doug Gwyn, Scott Anderson, Henry Spencer

HAMCu Installs a custom copper list for the current active view (usually workbench) that contains all the colours from 0x000 to 0xfff. A neat effect and an easy way to show off the color capabilities of the Amiga. Includes source. Author: Jonathan Potter

Image-Ed An shareware icon editor submitted by the author for inclusion in the library. Suggested shareware donation of $20. Version 1.8, binary only. Author: Jonathan Potter

JPClock A short clock program that is just packed with features. Includes source. Author: Jonathan Potter

MouseBounce A short hack/game that makes your mouse pointer bounce around the screen. The object is to close the MouseBounce window and exit the game. Each time you click the mouse button, the pointer speeds up. Includes source. by: Jonathan Potter

PopDir A small utility which "pops open" to help you look at the contents of a particular directory on demand. Version 1.4, includes source. by: Jonathan Potter

PopInfo A small utility which "pops open" to give you information about the status of your devices & memory. V 2.9, includes source. by: J. Potter

Teacher Teacher is a short, simple hack. I won't spoil the fun by telling you what it does. Includes source. Author: Jonathan Potter

Fred Fish Disk 205

Bally Amiga port of the former arcade game named Click. This version now has sound effects. Version II, an update to FF181. Binary only, shareware. by: Oliver Wagner

BattleForce A nicely done shareware game, submitted by the author, that simulates combat between two or more giant, robot-like machines. Binary only, version 3.01. Author: Ralph Reed

Chess A port of a chess game posted to Usenet. This is an update to the version first included on disk 96. It has been upgraded to use an Amiga Intuition interface. Version 2.0, binary only. Author: John Stanback; ported to Amiga by Bob Leivian Version 2.0 upgrades by Alfred Kaufmann

Fred Fish Disk 206

Brownian A demo based on both fractal theory and brownian motion. Includes source. Author: John M. Olsen

Hawk A stereo image of a hawk. Requires red/green stereo glasses to view. Author: Unknown (no documentation included)

MemFlick Treats all the memory in your Amiga like it was part of a bitplane inside a graphics display. Provides sort of a graphical picture of your memory usage. Binary only. by : Jim Webster

PeX A demo of the various graphics capabilities of the Amiga. Author: Unknown (no documentation included) PictureGarden Another demo, apparently in compiled basic. Author: Unknown (no documentation included)

StereoDemo A demo of stereoscopic graphics, written in assembly language. Requires red/green stereo glasses to view. Includes sources. Author: David M. McKinstry

Triple 3 demos of some of the Amiga's graphics and sound capabilities. Binary only. by: Tomas Rokicki

Fred Fish Disk 207

Coyote Gene's entry to the 1988 Badge Killer Demo contest. A very cute (and large) animation. Requires about 1900 blocks of disk space, so it is distributed in "arc format". Author: Gene Brawn

Fred Fish Disk 208

AsteroidField This is Michael's entry for the 1988 Badge Killer Demo Contest. It is a large animation of a spacecraft flying madly through an asteroid field (chased by unseen foes) that includes a couple of near misses. Author: Michael Powell

Fred Fish Disk 209

Bowl This is Vern's entry for the 1988 Badge Killer Demo Contest. It is a Sculpt-Animate animation that shows three colored balls flying in circles above a mirrored bowl. Rendering the animation took about 2 weeks. Distributed in zoo format because of its size (zoo program included for easy unpacking). Author: Vern Staats

Dps A program designed to work with the PrintScript program, a commercial PostScript interpreter for the Amiga, to provide a page previewer. V 1.1 and includes source. by: Allen Norskog

Fred Fish Disk 210

Calc A very nicely done scientific/programmer/plotter calculator. The scientific portion has most of the operations found on the more popular handhelds. The programmer portion has all the special hex/binary/decimal conversions as well as register operations like ASL, ROL, LSL, AND, OR, XOR, etc. The plotter portion will plot equations. Other features include 26 memories, full mouse or keyboard operation, pull-down menus, and iconization. V 3.0, binary only. by: Jimmy Yang

LabelPrint A program that allows you to easily print labels for your disks. Version 1.9, shareware, binary only (source available from author). Author: Andreas Krebs

NuHand An animation of a hand with fingernails scrapeing on a desktop, including sound effects. This is Bryan's entry for the 1988 Badge Killer Demo Contest. Binary only by: Bryan Carey Gallivan

Fred Fidh Disk 211

AmigaWave This is Allen's entry to the 1988 Badge Killer Demo Contest. It is an animation with sound effects by: Allen Hastings

Esperanto A keymap modification to usa1 which, in conjunction with the supplied slate.font, will allow one to type in Esperanto and Welsh, in any program that will use keymaps & fonts by: Glyn Gowing

Image-Ed An shareware icon editor submitted by the author for inclusion in the library. Suggested shareware donation of $20. V1.9, binary only. Fixes a serious bug in V1.8 on FF204. by: J. Potter

SignFont A keymap and font that will allow the user to be able to type in American Sign Language, provided that one knows the font. Author: Glyn Gowing

VirusControl A new virus detection and control program that checks disks during insertion, protects from link viruses, shows bootblock on a screen, periodically checks system vectors, controls access to files with a requester, etc. V1.3, includes full assembly language source code. Author: Pius Nippgen

Fred Fish Disk 212

Alice This animation is Carey's entry to the 1988 Badge Killer Demo Contest. Author: Carey T. Pelto

DiskSalv A disk recovery program for all Amiga file system devices that use either the Amiga Standard File System or the Amiga Fast File System. Disksalv creates a new filesystem structure on another device, with as much data salvaged from the original device as possible. Update to FF177. Binary only. Author: Dave Haynie

DogsWorld This animation is Charles' entry to the 1988 Badge Killer Demo Contest. by: Charles Voner

Fred Fish Disk 213

Cucug This animation of the Champaign-Urbana Commodore Users Group logo was submitted to the 1988 Badge Killer Demo Contest by Ed Serbe. by: Ed Serbe

Icons Almost 300 icons in eight (!) colors. Uses a special program to get an eight color workbench to display these icons, which were made with DPaintII and IconGen. Most icons are miniatures of the main screen of their corresponding programs, or the picture they show, made with "iconize" and "recolor" from FF85. by: Wolf-Peter Dehnick

Fred Fish Disk 214

ArcPrep	ArcPrep prepares files and/or directories for archival with arc or any other program that can't scan through different directories and/or handle long filenames. V2.1, includes source. Author: Garry Glendown
MandelVroom	A Mandelbrot/Julia-curve generating program that features five numerical generators (integer, ffp, ieee, 020, and 020/881) in hand-crafted assembly for maximum speed, online mouse selectable help for all functions, generation of multiple pictures simultaneously, a sophisticated user interface with shaded gadgets, etc. Some of the other features include zoom, magnify, color-cycling, contouring, auto-contouring, histogram, statistics, presets, extra-halfbrite support, overscan, orbits, pan mode, and more. Requires 1Mb or more of memory. This is the source to V2.0, an update to FF78. A compiled binary, along with help files and example images, can be found on FF215. Author: Kevin Clague
MemDiag	A memory diagnostic program to identify addresses which produce memory errors, and a memory quarantine program which removes such defective addresses from the system's free memory list, until such time as the hardware errors can be corrected. Version 1.1, includes source. by: Fabbian Dufoe
RunBack	Another step in the evolution of Rob Peck's RunBackGround program, from disks 73 and 152. Allows you to start a new CLI program and run it in the background, then closes the new CLI. This version has been enhanced to use the NULL: device by Gunnar Nordmark (included), which is a "real" device, so it solves problems with previous versions of RunBack which used the Nil: "fake" device, causing many crashes. Includes source. Author: Rob Peck, Daniel Barrett, Tim Maffett
SmartIcon	This shareware program, submitted by the author, is an Intuition objects iconifier. Version 1.0 is limited to iconifying windows, which is still very handy. It adds a new "iconify gadget" to each window, that when clicked on, iconifies the window into an icon in the ram: disk. This is the same version as released on FF134, but now includes the source code. Author: Gauthier Groult

Fred Fish Disk 215

MandelVroom	A Mandelbrot/Julia-curve generating program that features five numerical generators (integer, ffp, ieee, 020, and 020/881) in hand-crafted assembly for maximum speed, online mouse selectable help for all functions, generation of multiple pictures simultaneously, a sophisticated user interface with shaded gadgets, etc. Some of the other features include zoom, magnify, color-cycling, contouring, auto-contouring, histogram, statistics, presets, extra-halfbrite support, overscan, orbits, pan mode, and more. Requires 1Mb or more of memory. This is V2.0, an update to FF78. Source is available on FF214. by: Kevin Clague

Fred Fish Disk 216

BackDrop	Backdrop allows you to define a pattern which will then be displayed on the workbench screen in the normally empty area behind all the windows. Similar in concept to DropCloth, but this one does not require workbench to be loaded (and does not cohabit very well with workbench). Includes source. Author: Eddy Carroll
C64Emul	An April Fools spoof that turns your Amiga into a C64, or at least makes it look that way. Includes source. Author: Eddy Carroll
Cloud	A program that generates and displays fractal surfaces that look remarkably like clouds. Based on ideas from the book "Fractals" by Jens Feder. Binary only. Author: Mike Hall
PrtSpool	A DOS handler, a print program, and a control program that implement a print spooling system. Like PRT:, the DOS handler waits for stuff to be sent to it to be printed. The print program does line numbering and page headers. The control program handles administrative functions. Binary only. Author: Daniel Barrens
VirusX	Version 3.20 of the popular virus detection/ vaccination program. Features a test for 8 new viruses since the 3.10 version on FF175. Includes source. Author: Steve Tibbett
Wanderer	A neat little game with graphics and sound, ported from the Unix version, originally written on a Sun workstation. The idea for Wanderer came from games such as Boulderdash, Xor, and the Repton games from Superior Software. Includes a builtin editor for extending the game by adding additional screens. V2.2, includes source. Author: Steven Shipway and others. Amiga port by Alan Bland

Fred Fish Disk 217

AntiCBS	An animation cooked up by Leo in protest of CBS's coverage of the Hacker's Conference in Oct 88. After reading the transcript I was angered enough to feel this needed widespread distribution, even though it is quite old. by: Leo 'Bols Ewhac' Schwab
Echo	A small replacement for the AmigaDOS echo that will do some special functions, such as clear the screen, delete to bottom of screen, scroll the screen, place the cursor at a particular location, and set the text style and/or color. Includes source. Author: Garry Glendown
InstallBeep	This program replaces the DisplayBeep function so that an IFF 8SVX sound is played instead of the screen flashing. The PlayBeep function runs as a task in the background and runs asynchronously so the length of the sound does not slow anything down. Includes a couple of sample sound files. Version 1.1, binary only. Author: Tim Friest and Don Withey
SnipIt	An input handler wedge which allows you to clip text from any window and then paste that text anywhere, as though you had typed it on the keyboard. You mark the text you want to "snip" using the mouse, and then use the mouse to "paste" the last snipped text into the active window, requester, or anywhere. Version 1.2, includes source. Author: Scott Evernden
SonixPeek	A utility to let you list all the instruments used by one or more Aegis Sonix score files. It can scan individual files, or search one or more directories, checking all score files in each directory. The output is a list of all the instruments you need to have present in order to be able to play the indicated score files. Includes source. Author: Eddy Carroll
Stevie	A public domain clone of the UNIX 'vi' editor. Supports window-sizing, arrow keys, and the help key. V3.6, includes source. This is an update to V3.35a on FF197. Author: Various, Amiga work by G. R. (Fred) Walter

Fred Fish Disk 218

EdLib	A library of additional functions for Manx. This is V1.1, an update to V1.0 from FF183. Includes source. Author: Edwin Hoogerbeets with C-functions from several different authors
Mandel	Another mandelbrot generator program, with bits and pieces of code from C. Heath and R.J. Mical. This is V1.3, an update to FF111. New features and improvements include an ARexx interface, coordinates in sight, more state info saved with a picture, batch files, programmable functions, and more plotting options. Includes source. Author: Olaf Seibert
Maze	A program that lets you build mazes and then solve them. Mazes can be trivial one level mazes to very difficult three level mazes. Version 1.2, includes source. Author: Todd Lewis
PcPatch	Patches for PCCopy and PCFormat from the EXTRAS disk, to allow reading, writing, and formatting of any kind of MS-Dos style disks, including 720K 3.5" diskettes. Binary only. This is an update to the version on disk 163. Author: Werner Guenther
Scanner	Scanner makes commented C code of all intuition structures in memory. The structures will receive correct pointers towards each other. Scanner starts looking at IntuitionBase, and follows all pointers, storing them in memory. When finished, it writes all the structures to the standard output. Version 1.0, includes source. Author: Stefan Parmark
Worm	An Amiga implementation of the classic "worms" program, based on an article in the Dec 1987 issue of Scientific American. You can specify the size and length of the worms, and the number of worms. Includes source. Author: Brad Taylor, Amiga port by Chuck McManis

Fred Fish Disk 219

DeepSky	A database containing information on 10,368 non-stellar objects, 600 color contrasting easily resolved double stars, 70 stars for setting circles, and misc white dwarfs, red stars, binaries, etc. The database is distributed in zoo format, and is about 1.2 Mb after extraction. V 5.0. Author: Saguaro Astronomy Club
Mv	A Unix style mv/cp/rm program that moves, copies, or removes files. Includes interactive mode, recursive mode, and force quiet mode. Copies file permissions, dates, and comments, supports arp style wildcards, supports moves across volumes, honors the delete bit. V 1.1, includes source. Author: Edwin Hoogerbeets

Fred Fish Disk 220

DNet	A link protocol that provides essentially an unlimited number of reliable connections between processes on two machines, where each end of the link can be either an Amiga or a Unix (BSD4.3) machine. Works on the Amiga with any EXEC device that looks like the serial.device. Works on UNIX with tty and socket devices. Achieves better than 95% average throughput on file transfers. This is V2.0, an update to FF145. Includes sources for both the Amiga and Unix versions. Author: Matt Dillon

Fred Fish Disk 221

AllocMaster	Allocmaster is a program inspired by Nick Sullivan's "Reserve" article in Amiga Transactor, for controlling the amount of both Chip and Fast memory available to the rest of the system. It is very useful for testing applications in low-memory situations. It also has a snapshot feature to report differences in available memory before and after running an application. V 1.17, binary only. Author: John Gerlach Jr.
ANSIEd	Demo version of an ANSI screen file editor. It allows you to easily create and modify a screen of ANSI-style text/graphics on the Amiga. The standard ANSI color set (red, green, yellow, blue, magenta, cyan, white) and text styles (plain, boldface, underlined, italic) are provided, along with some simple editing and drawing functions. This demo version has the save features disabled. V1.2.0aD, binary only. Author: Gregory Epley
BallyII	Amiga port of the former arcade game named Click. This version adds a "cheat" mode and fixes some minor bugs. VII+, an update to FF205. Binary only, shareware.Author: Oliver Wagner
DFrame	A utility that helps you to create animated bobs. It installs itself in DPaint II, after which you can draw each bob in DPaint II within its own frame and check the animation by calling DFrame from within DPaint. V1.02, binary only.Author: Jan Buitenhuis
IFFM2	Demo version of an IFF support module for Interface Technologies M2Amiga Modula-2 system. Includes a version of ViewILBM (with source) that uses the IFF support routines. V1.0.0D, binary only. Author: Gregory Epley
Steinschlag	A tetris like game (Steinschlag means "Falling Rock") submitted by the author. V1.5, binary only. Author: Peter Handel

Fred Fish Disk 222

MemGauge	A tool to display the current memory usage, very much like the usage bar Workbench displays in root directories. V1.4, includes source. Author: Olaf 'Olsen' Barthel
Mischief	This little program is in the long tradition of "display hacks". It uses the input.device to perform various acts of mischief. Includes source. Author: Olaf 'Olsen' Barthel
Plplot	A library of C functions useful for scientific plotting on the Amiga. The library is Lattice C compatible. Contour plotting, three dimensional plotting, axis redefinition, log-log plotting and multiple subpages are a few of Plplot's features. The plots can be displayed on a monitor or sent to a graphics file for subsequent printing. V1.00, includes source. Author: Tony Richardson

Fred Fish Disk 223

Csh	V3.03a of a csh like shell derived from Matt Dillon's shell, version 2.07. This is an update to FF199. Includes a couple of new filter commands, new dir option, new editing options, sourcing of a standard startup file, and some bug fixes. Includes source. Author: Matt Dillon, Steve Drew, Carlo Borreo, Cesare Dieni
FixDisk	A program to recover as much as possible from a defective disk. It can sometimes recover damaged (unreadable) tracks, check file integrity, check the directory structure, undelete files, copy or show files, fix corrupted directory pointers, etc. Full intuition interface. Version 1.0, binary only. Author: Werner Guenther
GravSim	A program to animate up to 6 planetary masses, all of which exert a mutual gravitational force on each other. The planetary masses can be placed anywhere on the screen, and their mass and initial velocity can be determined by the user. The program then steps the animation through time, plotting and displaying the new position in the trajectory of each mass. V1.50, includes source. Author: Richard Frost
Iff2Sun	A small utility for those of you who may have access to a Sun workstation. Takes an Amiga IFF file and converts it to a Sun rasterfile format. Update to FF174, with better parsing, support for HAM mode, and some bug fixes. Source only, as the program needs to be re-compiled and run from a Sun. Authors: Steve Berry, Mark Thompson
IFFtoSUN	This program takes a standard IFF format image and translates it into a SUN rasterfile format, like the Iff2Sun program also on this disk. However, this one runs on the Amiga. V1.31, includes source. Author: Richard Frost
Paccer	A pacman clone with sound and a game screen editor. V1.0, shareware, binary only. Author: Dirk Hoffman
PopInfo	A small utility which "pops open" to give you information about the status of your devices and memory. V3.0, an update to FF204. Includes source. Author: Jonathan Potter
SetCPU	A program designed to allow the user to detect and modify various parameters related to 32 bit CPUs. Includes commands to enable or disable the text/ data caches, switch on or off the '030 burst cache line fill request, use the MMU to run a ROM image from 32-bit memory, and to report various parameters when called from a script. V1.5, an update to FF187. Includes source. Author: Dave Haynie

Fred Fish Disk 224

CLImax	For all those people who wish that their CLI windows had 25 lines of 80 characters just like an old fashioned non- windowing computer, the answer is here. CLImax creates a borderless backdrop CLI window on a custom screen. Also thrown in is MoveSys, which reassigns SYS:, C:, S:, L:, DEVS:, LIBS:, and FONTS: to a new volume with one simple "pure" command. Includes source. Author: Paul Kienitz
KickMem	A program for A1000 hardware hackers that have done the Amazing Computing 512K upgrade. KickMem will patch your 1.2 or 1.3 kickstart disk to perform addmem during kickstart. This allows warm boot surviveability of ram disk devices and eliminates addmem commands from your startup sequence. V2.0, includes source. Author: Dave Williams
MoreIsBetter	These two hacks make MORE more useful. One is called V; it's a small "pure" CLI command that acts as a front end for More, causing it to create its own window. Make V and More both resident! The other is Fenestrate, which surgically alters the CON: window spec inside More enabling it to, for instance, use ConMan features to create a borderless window on the topmost screen (very useful with CLImax). Includes source. Author: Paul Kienitz
PetersQuest	This cute game has you, the intrepid Peter, following a trail of hearts through a world of 20 levels, riddled with porcupines and other hazards, to rescue Daphne, the love of your life that has been kidnapped by the evil Brutus. Version 1.0, binary only. Author: David Meny
Who	A rewrite of "who", from FF79, which gives substantially more elaborate information about the tasks currently running (or waiting) on your Amiga. Includes source. Author: George Musser, rewrite by Paul Kienitz
Xebec	A couple of hacks to make life easier for those who have Xebec hard disks. One makes it more possible to Mount a Xebec hard disk with the Fast File System, the other is a compact head parking program. Includes source. Author: Paul Kienitz

Fred Fish Disk 225

AmigaTCP	The KA9Q Internet Software Package. The package supports IP, ICMP, TCP, UDP, and ARP as basic services, and implements the FTP, Telnet, and SMTP protocols as applications. It runs on IBM PC and clones, the Apple Macintosh, and the Amiga. Includes source. Author: Bdale Garbee, Phil Karn, Brian Lloyd
MyMenu	MyMenu allows you to add your own menus to the WorkBench menu strip, to run commonly used commands. MyMenu will allow you to execute both CLI and WorkBench programs, and is configured with a normal text file. Includes source. Author: Darin Johnson

Fred Fsih Disk 226

Vlt	VLT is both a VT100 emulator and a Tektronix (4014 plus subset of 4105) emulator, currently in use at SLAC (Stanford Linear Accelerator Center). Although the VT100 part was originally based on Dave Wecker et al.'s VT100, many enhancements were made. The program requires ARP, and it has an ARexx port. XMODEM 1K/CRC and Kermit protocol support also included. V4.036, with many enhancements over the previous version, 3.656, on FF202. New features include support for other serial ports, external file transfer protocols, and "chat" mode. Improved behavior on the Workbench. Tektronix emulation now allows saving IFF files, PostScript files, and printing bitmaps to the printer. Many other enhancements and bug fixes. Binary only. Author: Willy Langeveld

Fred Fish Disk 227

MidiLib	A disk based library that permits sharing of the serial port by MIDI applications through a MIDI message routing and processing system. The midi utilities include a midi monitor to display incoming midi messages to the console, a routing utility, a midi library status utility, and more. V2.0, an update to FF101, and includes significant speed enhancements, new utilities to play with MIDI files, and updated utilities, documentation and examples. Binary only (source for examples and bindings however). Author: Bill Barton
PickPacket	PickPacket gives you a visual display of the DosPacket structures that are sent to handlers, and lets you see the results. You can actually perform handler operations such as open files, read or write data, Examine or ExNext locks, and so forth, all by talking directly to the file system handler involved using PickPacket. V1.0, includes source. Author: John Toebes and Doug Walker
RexxArpLib	A library which originally was supposed to be an ARexx interface to the ARP library. However, it has also become an interface to various Intuition functions, containing over 50 functions including a file requester, string/boolean requester, environment variable functions, simple message window, wildcard expander, etc. V2.3, an update to FF178. Binary only. Author: W.G.J. Langeveld
RexxMathLib	A library which makes various high level math functions such as sin, tangent, log, etc, available in ARexx. V1.2 and 1.3, binary only. Author: W.G.J. Langeveld

Fred Fish Disk 228

Az	A nice little text editor that is fast, simple to use, and very Amiga'ized. V1.40, binary only.Author: Jean-Michel Forgeas
Glib	A text screen oriented librarian and editor for synths. Supports the TX81Z, DX100, DEP5, DW8000, and K-5. Includes source. Author: Tim Thompson, Steve Falco, and Alan Bland
JazzBench	A drop-in multitasking replacement for WorkBench. It has more features that WorkBench and is fully multitasking (no more waiting for ZZZ clouds). It allows you to extend it, add your own menus, key shortcuts, etc. This is alpha version 0.8, binary only. Author: David Navas
Xoper	Very comprehensive program to monitor and control system activity. Monitor cpu, memory usage, ports, interrupts, devices. Close windows, screens, show loaded fonts or last Guru code number. Clean up memory, flush unused libraries, devices, fonts. etc. and a whole bunch more! Spawns its own process. A very handy background task to have loaded. V1.3, an update to FF171. Assembly source included. Author: Werner Gunther

To Be Continued......

In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the public domain by their authors, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the authors' wishes, please contact us by mail.

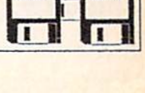
IMPORTANT NOTICE!

This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial groups only! Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.

Any non-commercial Amiga user group wishing to duplicate this list should contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

PiM Publications Inc. is extremely interested in helping any Amiga user groups in non-commercial support for the Amiga.



For PDS orders, please use form on page CIII
Visa and MasterCard is available on orders of $20.00 or more.

Amazing Deal!

Name ______________________________

Address ______________________________

City ______________________ State ____________ Zip ____________

Charge my ☐Visa ☐MC # ______________________ Expiration Date ____________ Signature ____________

All Charges are subject to a $20.00 minimum

dial 1-800-345-3360

PROPER ADDRESS REQUIRED. In order to expedite and guarantee your order, all large Public Domain Software orders, as well as most Back issue orders, are shipped by United Parcel Service. UPS requires that all packages be addressed to a street address for correct delivery.

One Year Of AC! Please circle the appropriate item: **New Subscription** **Renewal**

Our regular renewal now includes 12 monthly issues of **Amazing Computing** **plus** the Spring edition of *AC's GUIDE/AMIGA*

☐ $28.00 U.S ☐ $44.00 Foreign Surface ☐ $36.00 Canada and Mexico

(air mail rates available on request)

Our New SuperSub!

12 monthly issues of **Amazing Computing** **plus** *AC's GUIDE/AMIGA*

3 Product Guides! Spring, Fall, and Winter! A savings of $32.25 off the newsstand price.

Please remember, the Product Guides alone retail for $6.95 each!

☐ $36.00 U.S ☐ $52.00 Foreign Surface ☐ $44.00 Canada and Mexico

(air mail rates available on request)

Back Issues: $5.00 each US, $6.00 each Canada and Mexico, $7.00 each Foreign Surface.

Please circle your Back Issue choices below:

Vol1.1 Vol1.2 Vol1.3 Vol1.4 Vol1.5 Vol1.6 Vol1.7 Vol1.8 Vol1.9 Vol2.1 Vol2.2
Vol2.3 Vol2.4 Vol2.5 Vol2.6 Vol2.7 Vol2.8 Vol2.9 Vol2.10 Vol2.11 Vol2.12 Vol3.1
Vol3.2 Vol3.3 Vol3.4 Vol3.5 Vol3.6 Vol3.7 Vol3.8 Vol3.9 Vol3.10 Vol3.11 Vol3.12
Vol4.1 Vol4.2 Vol4.3 Vol4.4 Vol4.5 Vol4.6 Vol4.7 Vol4.8

Public Domain Software:

$6.00 each for subscribers (yes, even the new ones!)

$7.00 each for non subscribers **(three disk minimum on all foreign orders)**

Please circle your Public Domain Software choices below:

Amazing on Disk: A#1. . .Source & Listings V3.8& V3.9 A#2. . .Source & Listings V4.4
A#3. . .Source & Listings V4.5 & V4.6 A#4. . .Source & Listings V4.7 & V4.8

InNOCKulation Disk: IN#1...Virus protection

AMICUS 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26

Fred Fish Disks

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 NA
58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76
77 78 79 NA 81 82 83 84 85 86 87 NA 89 90 91 92 93 94 95
96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114
115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133
134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152
153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171
172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190
191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209
210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228
229 230 231 232 233 234 235 236 (NA Denotes disks removed from the collection)

Please complete this form and mail with check, money order or credit card information to:

PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722-0869

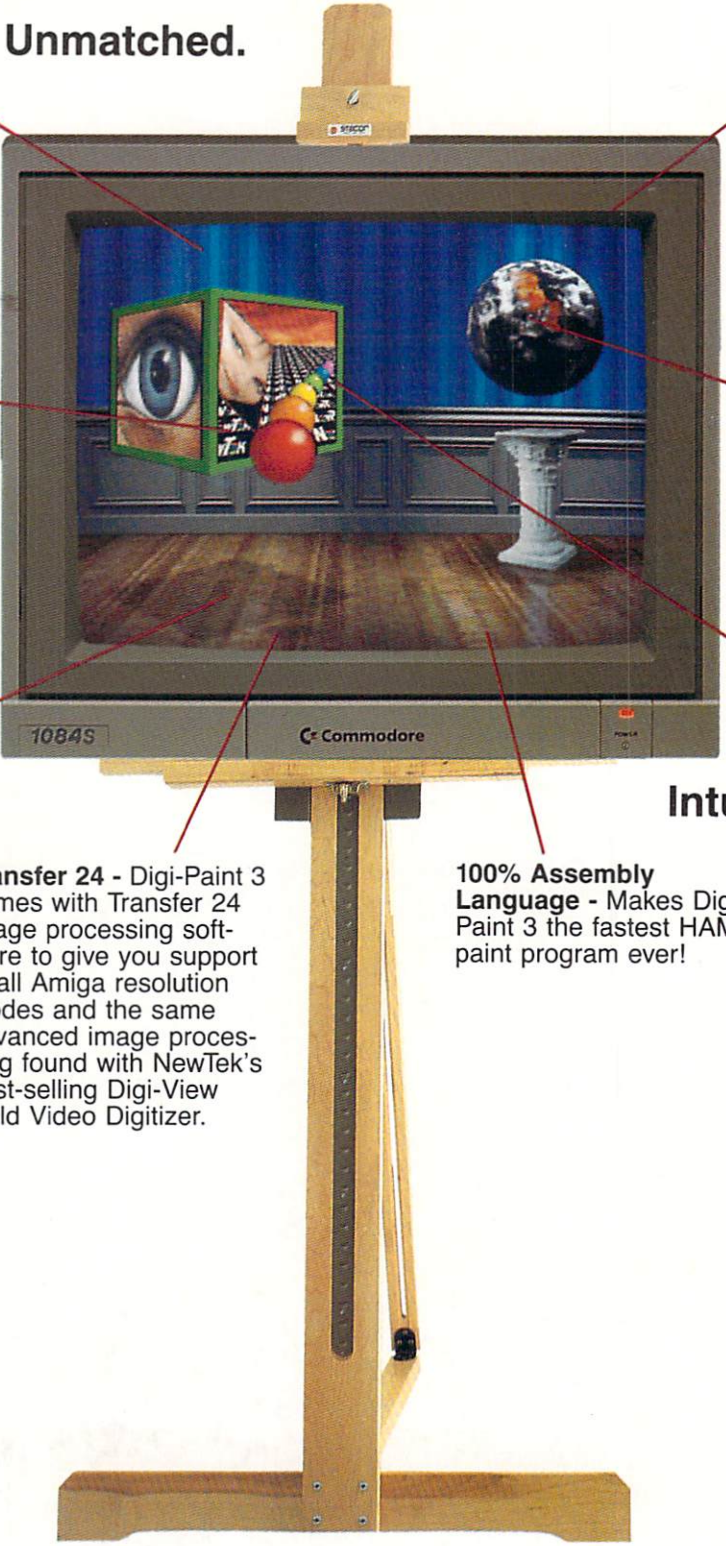
Please allow 4 to 6 weeks for delivery

Subscription: $ __________

Back Issues: $ __________

PDS Disks: $ __________

Total: $ __________



Variable Dither - Computed internally at 30 bits per pixel (over one billion colors). Gives you over 100,000 apparent colors on screen.

Sophisticated.

Flexible Text Rendering - Allows for anti-aliased fonts, Rainbow Fonts and Transparent Fonts and more.

Revolutionary.

User-Controllable Transparency - Allows real time control of the amount of transparency and the location of the light source.

Unlimited.

Unmatched.

Transfer 24 - Digi-Paint 3 comes with Transfer 24 image processing software to give you support of all Amiga resolution modes and the same advanced image processing found with NewTek's best-selling Digi-View Gold Video Digitizer.

100% Assembly Language - Makes Digi-Paint 3 the fastest HAM paint program ever!

Super BitMaps with Auto-Scrolling - Real-time scrolling on up to 1024 pixels high or wide image with full overscan display.

Flexible.

Colorize - Play Ted Turner and add color to black-and-white images or change colors on already colored images.

Powerful.

Texture Mapping with Anti-Aliasing - Gives you super-fast warping and stretching of any image.

Intuitive.

The Ultimate Paint Program:
DIGI·PAINT™ 3

For more information call NewTek at 800-843-8934 or 913-354-1146

Digi-Paint 3, Digi-View Gold and Transfer 24 are trademarks of NewTek Inc.

NEWTEK
INCORPORATED

Circle Reader Service Card No. 102